Ministry of transport and telecommunications of the Ukraine
State Department concerning telecommunications and informatization

ODESSA NATIONAL ACADEMY
OF TELECOMMUNICATION named after A.S.POPOV

Department of the felecommunication theory named after A.G. Zjuko

TELECOMMUNICATION THEORY

MODULE-4

ERROR-CORRECTING CODING in TELECOMMUNICATION SYSTEMS

Manual for students,
trained to a higher education direction
6.050903-Telecommunications

Odessa 2010

УДК 519.95

ББК 318

Composed by prof. **Victor Banket**

V.Banket. Error-correcting Coding in Telecommubications. Manual for students. Module-4 on discipline «Telecommunication Theory» – Одесса:

ОНАС, 2010 – 78 p.

Manual contains the materials of the Module-4 discipline «Telecommunication Theory», presented in the form of lectures. Each lecture is accompanied by references, questions and tasks for students work.

Content

Introduction

In modern telecommunication systems the information, as a rule, is transferred in a digital form and processed by a digital methods. Thus the important role is played *the coding information methods*.

The teaching material is broken into the numbered Lectures. The Lectures are devoted a statement of the error-correcting coding theory. The theoretical material is explained by Examples and Exercises.

At the beginning of each Lecture the lecture plan is resulted, in which references to the recommended literature are given. For checking of knowledges it is recommended to use the Questions placed in the end of each Lecture. In the end of Lecture for students independent work the Tasks for the decision of problems are given. Tasks and methodical instructions for the making of Course Work on the Module-4 subjects are given in the Attachment B.1. In Attachment A help data about code characteristics are given.

For the best mastering of a theoretical Lectures it is recommended to familiarise preliminary with the text of each Lecture and to prepare questions to the lecturer.
After each Lecture to make the personal abstract with using of the present manual, the literature and to make written answers to control Questions. In the end of each lecture are the Tasks having the form of problems. Some Tasks have the raised difficulty (are noted by a sign*). They can be solved in an auditory with the teacher on a practical training.

> The most important terms and theory positions are italicised also *by special frames.*

The theoretical material of Module-4 is stated in the form of the numbered Lectures (L-1 … L-13). For the convenience the double numbering of a each Lecture elements (formulas, figures, tables, examples, tasks, questions and exercises) is accepted. The first figure of number specifies Lecture's number and the second the serial number – an element within Lecture. For example, numbers (1.2) and Figure 1.3 designate numbers of the formula and figure from the first Lecture.

Lecture 1
Appointment, structure and classification of error-correcting codes

Plan

1.1. Error-correcting codes in telecommunication systems [1, Section 10]
1.2. Classification of an error -correcting codes [1, Section 10.1]

1.1.Error-correcting codes in telecommunication systems

In the theory of a modern telecommunication systems the considerable attention to *coding methods* of the information is given.

> *Coding* – an operation of an identification of the symbols or groups symbols from one code by symbols or groups symbols to other code. Necessity of coding arises, first of all, from requirement to adapt the message form to the given communication channel or to any other device intended for transformation or storage of the information.

The typical block diagramme of digital telecommunication system is resulted on figure 1.1. The source produces messages which it is necessary to transfer through the channel of a telecommunication systems. It can be sequences of *discrete messages* (data, cable messages etc.) or *continuous messages* (speech, audio, TV, etc.), transformed to *the digital form.*

The real messages contain *redundancy* and for the matching of the information source with the transmiting channel usually use *the source encoder.* Together with *the decoder* they form *the source codec.* The source coding methods were studied in the Module-2. The primary goal of the any telecommunication system is *the information transmitting with the given fidelity and speed.* These requirements are in contradict, and, increasing of information speed leads to decreasing of the noise immunity and transmitting fidelity.

In the agree with the well known Shannon theorems, as is wished considerable increase of fidelity of the information transfer if a transmiting rate through the channel R_{chan} does not exceed the channel *capacity* C basically is possible. It is reached by using of the enough long *error-correcting codes* (ECC).

> *The error-correcting codes are* the codes which allow to *detect and correct of the errors* arising from noises and distortions in the communication channels.

With that end in view the *redundancy* is entered into structure of error- correcting codes. *Codec ECC* (*the channel encoder and the decoder*) are shown on figure 1.1. In real conditions the length of a code is limited by admissible *complexity* of coding and, first of all, for decoding devices.

Therefore the result from the using of error-correcting codes depends on the parametres of a code and restrictions on realisation of the channel codec.

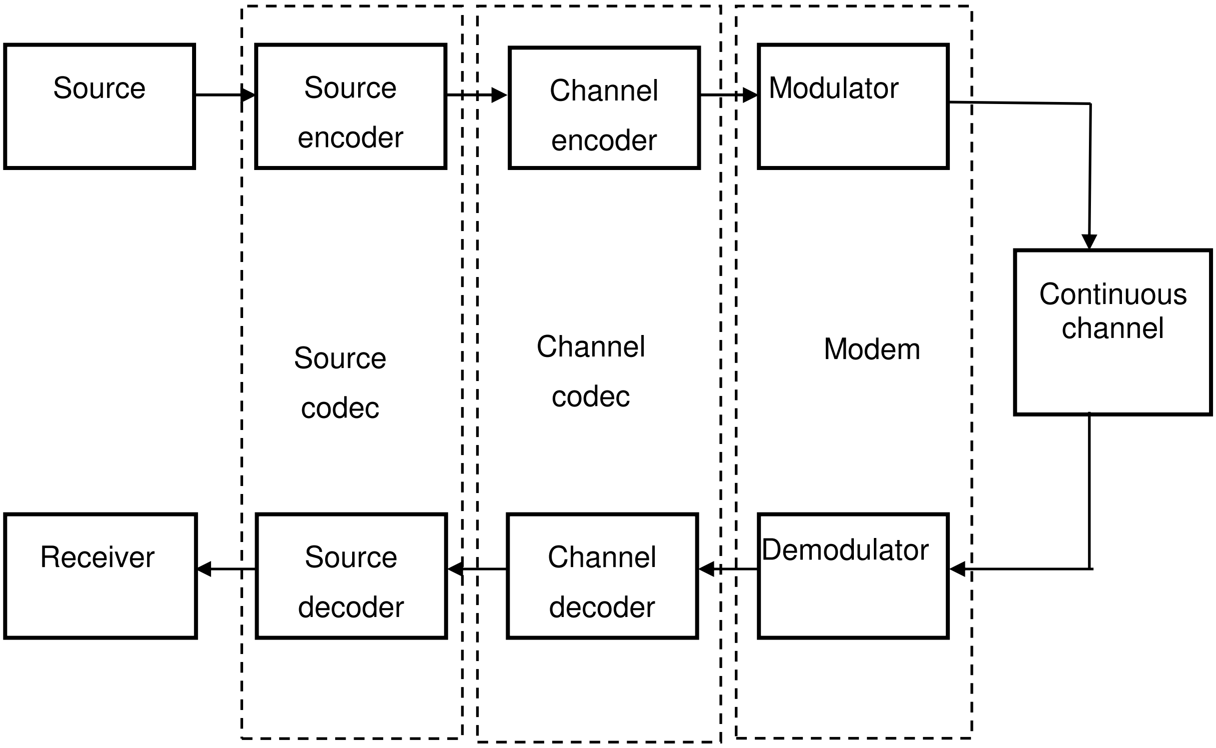
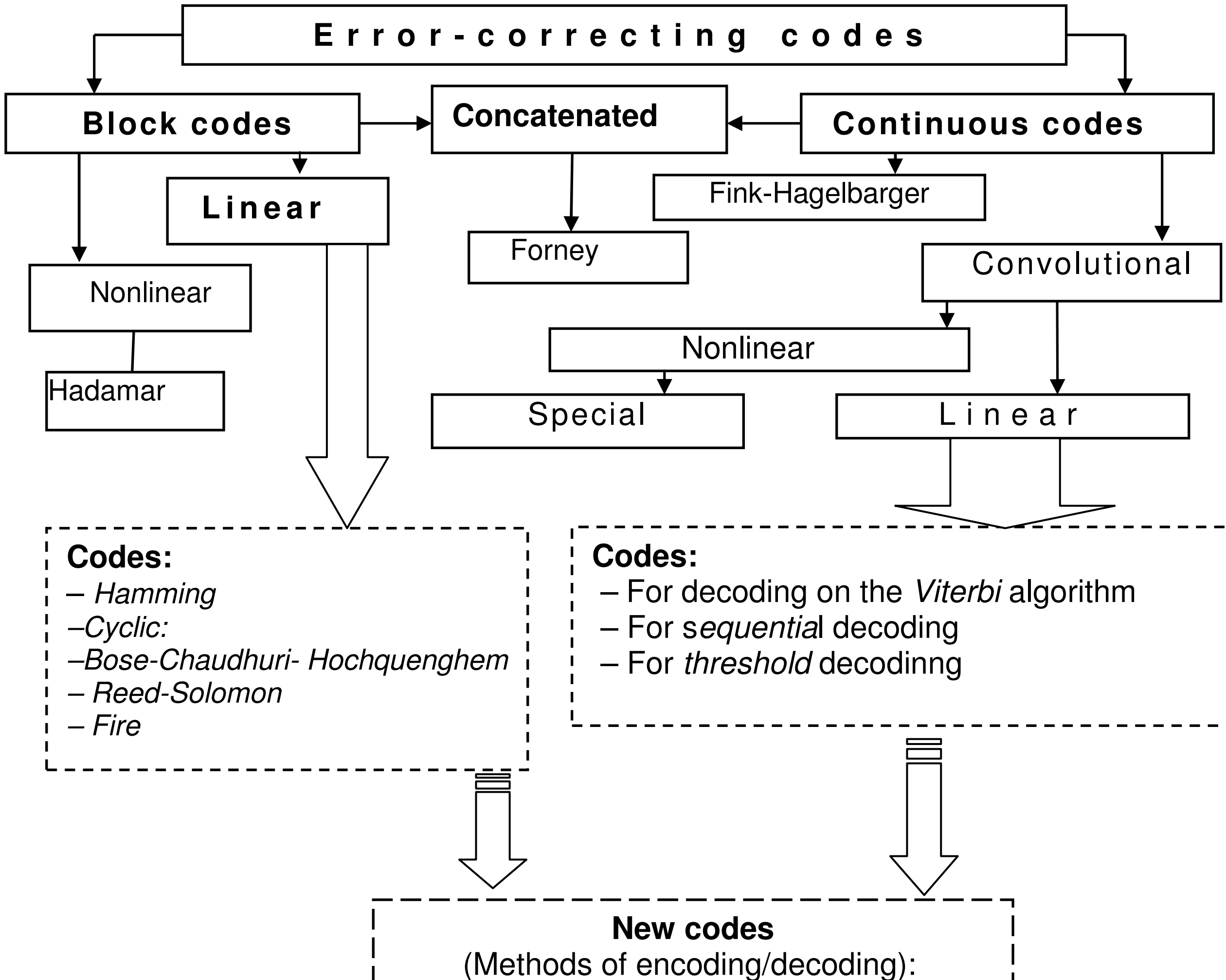


Figure 1.1–Typical block diagramme of a digital telecommunication system

The modern theory offers a wide set of error-correcting codes, various on structure, construction principles and error-correcting ability. In the subsequent lectures the important classes of the codes with effective coding/decoding algorithms are considered.

1.2. Classification of the error-correcting codes

The error-correcting codes can be classified to various signs. The structure of classification of codes is resulted on figure 1.2. On a way of formation ECC are subdivided on *block* and *continuous* codes. Formation of the *block codes* provides *splitting* of transferred digital sequences *into separate blocks* which move to a encoder input. To each such block on an encoder output there corresponds the block of the code symbols which work is defined by a rule named as the *coding algorithm.* Formation of the *continuous codes* is carried out *continuously in time, without division into blocks* as defines the name of this class of codes. Block codes historically have been discovered and studied earlier, at the beginning of development of the coding theory. In a class of continuous codes it is necessary to note *a convolutional codes* which exceed on characteristics of the block codes, and, for this reason, find wide application in a telecommunication systems. Many codes carry names of scientists which have discovered and investigated them. Such examples are the continuous Fink-Hagelbarger's code offered by Soviet scientists L.M.Fink and German expert R. Hagelbarger. Long time this code was in the literature as an indicative example of a continuous code with simple encoding/decoding algorithms, but after elaborating a *convolutional codes* has given way to them. For the description of procedures of coding/decoding both block, and convolutional codes usually use an *adequate mathematical apparatus.* For the description of *linear codes* the well developed linear algebra is used. Formation of *nonlinear codes* is made with application of nonlinear procedures. Such approach allows to construct in some cases *nonlinear codes* with a number of special properties. In the error-correcting coding theory the problem of *realisation complexity* encoding/decoding procedures and in particular decoding procedures is important. Therefore some classes of codes (*Hamming codes*, *Bose-Chaudhuri-Hochquenghem codes, Reed-Solomon codes, Fire codes* etc.) have been developed together with the decoding algorithms connected with structural properties of these codes. And, on the contrary, the elaborating of a new decoding algorithms for convolutional codes (*Viterbi algorithm*, *sequential decoding, threshold decoding)* initiated a searches of the corresponding codes. Distinctive advantages of error-correcting codes induced searches of new approaches to realisation of ways to increase of a noise immunity and efficiency of telecommunication systems. On figure 1.2 *new methods* of encoding/decoding(*signal-to-code structures, turbo-codes,time-space coding)* are noted accordingly.



Questions

1.1.What is an appointment of the error-correcting coding by transmitting of the digital information?
1.2. The codec of a error-correcting code consists of what elements?
1.3. In what difference of coding procedures the block and continuous codes consists?

Tasks

1.1. Represent the block diagramme of telecommunication system and describe appointment of its separate blocks.
1.2. Result classification of error-correcting codes by ways of formation and structural properties.
1.3. Result the scheme of inclusion of the encoder and decoder of the error-correcting code as a part of digital telecommunication system. Explain appointment of the scheme elements.

Lecture 2

Parametres of the block error-correcting codes

Plan

2.1. Key parametres of the block error-correcting codes [1, Section 10]

2.1. Key parametres of the block error-correcting codes

There are the following *parametres* of the block codes.

The *code basis m* is the number of the various symbols used by a coding.

In practice a codes with the basis *m*=2 are used. These are *binary codes***.** For construction of binary code word *the binary alphabet* with symbols {0,1} is used. Wide practical using of binary codes is defined for a reason of simplicity of a binary logic elements construction in the codec memory devices. A block code consists of a set of fixed length vectors named *code words.* The length of a code word is the number of elements in the vector and is denoted by *n.*

> The *length* of *a code word n* is the number of elements in the code vector**.**

Redundancy in the block code words can be entered as follows.

Let on a block encoder input *the information block* ***a*** = {a_1, a_2, a_3, a_4, …, a_k} arrives. By the block coding *code word* on the encoder exit can look like:

$$\boldsymbol{b}=\{a_1,a_2,a_3,a_4,\dots,a_k,\ c_1,c_2,c_3,\dots,c_{n-k}\},$$

where (c_1, c_2, c_3, …, c_{n-k}) – *additional symbols*. Values of additional symbols are defined by *coding rules***.** Such code is called as *systematic.*

> Each code word of length *n* symbols contains in a systematic codes *k* information symbols. Thus to an information symbols are added *r=(n–k)* additional symbols which are depend from information symbols and used by the decoding *for detection and correction* of an errors. *In nonsystematic codes* information symbols *in an explicit form* in a code word *do not contain.*

The total quantity of the *possible* code words of the block error - correcting code is defined by the formula:

$$M=m^n. \tag{1.1}$$

For a possibility of detection and correction of an errors these *M* code words *not completely use* for an information transfer. From these m^n code words we may select $M_0=m^k$ code words ($k<n$) to the forming a code. Thus block of *k* information bits is mapped into a code word of length *n* selected from the set of $M_0=m^k$ code words. These *words* named *allowed* as they are allowed for an information transfer.

We refer the resulting *block code* as an*(n,k) code*, and the ratio

$$R_c=\frac{k}{n} \tag{1.2}$$

is defined to be the code rate **.**

Quantity of the allowed code words is equal

$$M_0=m^k.$$

> In the error-correcting code possible words are used *not completely* i.e. $M_0<M$**.** It illustrates *redundancy* of a code.

The rate of a error-correcting code is defined also by the ratio

$$R_{code}=(\log_m M_0)/(\log_m M) \qquad (1.3)$$

In nonredundancy codes $M_0=M$ (or $k=n$) and the rate is

$$R_{code}=1 \qquad (1.4)$$

For the characteristic of error-correcting codes enter concept *redundancy* of a code.

Redundancy of a systematic codes χ_{red} is *a relative share* of the number of additional symbols $(n-k)$ in a code word on its length n symbols:

$$\chi_{\text{red}}=1-R_{\text{c}}=(n-k)/n \qquad (1.5)$$

For simple (*nonredundancy*) codes ($n=k$) $\chi_{\text{red}}=0$.

Exercise 2.1. As is known, in a binary channels under the noises and distortions there are an errors in the form of transitions of a transferred symbols to an opposite symbols. For example, by transfer of a symbol 1 transition (1→0) is possible and, accordingly, transitions (0→1) are possible also.Consider the possibilities of construction of the binary error-correcting code intended for transfer of messages with symbols from alphabet with volume of M_A, and allowing by the receiving *to detect* the channel errors. Specify the encoding and decoding methods of such code. For the developed algorithm of a coding define the rate and redundancy of such ECC.

Instructions. The providing of an errors detection in the transmitted code words will be possible if for the allowed code words to give a forms which changes by errors in symbols of this words. Then detection of errors (i.d. decoding) can be made by*check of conformity* of the received words to this in advance known forms. At the first development times of the error-detecting codes the maintenance in the transmitted allowed words *of «even number of unit symbols»* was considered as simple way. So the «*Code with even number of units* » has been invented.

Decision. We will consider a variant of construction of the binary systematic code intended for transfer of letters, chosen from the alphabet of a volume M_A. According to above considered rule the information block $\boldsymbol{a}=\{a_1, a_2, a_3, a_4, \ldots$ and, $a_k\}$ of each word should contain k binary symbols a_i. The total quantity of information blocks should be precisely equal to volume of the source alphabet M_A. That is the equality $M_A.=2^k$ guarantees transfer of each symbol of the source, and the corresponding to it code words of a systematic code. The quantity of units in an information blocks depends from a primary simple code and can be both even and odd. It appears that for realisation of encoding and decoding of such code words it is convenient to use the procedure «*module-2 addition*», defined in the manual [3, Section 2.1.2]. This procedure defines the simple way to find of the *parity of units number* in a code word. To everyone information block we will attribute one *additional symbol* (r=1) so that the quantity of units in again formed word was *even.*

*Encoding i*t is made in such sequence:

1. Let *information block* ***a*** is represented by a primary code: $\boldsymbol{a}_1$→101010;
2. By consecutive module-2 addition of the primary code symbols defines an additional symbol c=1;

3. We form allowed code words, finishing an additional symbol to the block of information symbols ***b***=1010101. It is visible, that the coding rule is carried out, since the number of units remains even;
4. By the other form of a primary code it is received: $\boldsymbol{a_2}$→101011, *c*=0 and $\boldsymbol{b}_2$=1010110.
5. It is obvious, that any transition ((1→0) or (0→1)) changes number of units in the received words. If by decoding to use procedure of calculation of units number it is possible to detect errors.

Remark. It appears, such code allows *to* detect *not any errors configurations.* The simple analysis shows, that two-multiple change of symbols cannot change parity and such errors in this code to *detect it is impossible*. It is recommended to make such analysis for other variants of error combinations independently.

The *rate* and *redundancy* of a code with even units number and by parametres : *k*, *r*=1*, n*=*k*+*r*=*k*+1 are defined by formulas:

$$R_{code}=\frac{k}{n}=\frac{k}{k+1}\text{ and }\chi_{red}=\frac{n-k}{n}=\frac{1}{k+1}.$$

It is visible, that for the big lengths of the information block *k>>1* the rate of such code is close to R$_{code}$=1, and redundancy by transfer for example letters from the Russian text with alphabet volume M_A. = 32 (*k*=5*)* will be small $\chi_{red}=\frac{1}{6}$.

Questions

2.1.What is the reason of wide application of binary codes in telecommunication systems?

2.2.Make definition of the systematic block code.

2.3.Whether placing of additional symbols in front of the block of information symbols in a systematic code is possible? Whether will change it redundancy of a code?

Lecture 3

Block codes ability to detect and correct of the errors

Plan

3.1. Code ability to detect and correct of the errors [1]

Instruction. Study the elements of the general algebra from the manual [3, Section 2]

3.1.Code ability to detect and correct of the errors

Let's establish dependence of *detecting and error-correcting ability* of the block codes from a code parametres. It is useful to consider a binary code with parametres n=3, k=2. All words of this code (M=8). It is possible to divide by sign *«parity of units number in a code words»* on two groups:

–Words with *even* number of units,

– Words with *odd* number of units.

The code constructed by this principle named "*A code with even number of units*" is considered in the Exercise 2.1.

Example 3.1. A binary code(m=2, n=3) with even number of units.

In table 3.1 the full set of binary words (m=2,n=3, M=8) is divided into a set of the *allowed* code words (M_0=4) containing words with even number of units (including a word 000 (number 0 – even)), and the set of the *forbidden* words with odd number of units. Their total quantity is equal to difference $M_{forbid.}$=M– M_0=4.

The *allowed code words* are used for an information transfer through the channel (Are *allowed* for transfer).

The *forbidden* words are not used for an information transfer through the channel (Are *forbidden* for transfer).

Full set of a words (M=*8):* {000,001,010,011,100,101,110,111}	
The *allowed* code words (*with even number of units)* M_0=4 {000,011,101,110}	The *forbidden* words (*with odd number of units)* $M_{forbid.}$=M–M_0=4 {001,010,100,111}
Code parametres: Code rate R_{code}=1/2, Code distance d_{min}=2, Code can detect $q_{\det}$=1 errors	

Table 3.1 – Code with even number of units

In the coding theory the important role plays the concept *distance between code words.* Everyone binary block error-correcting code are characterised by a parametre *code distance.*

The *code distance d_{min} is* the minimal *Hamming distance* between the allowed code words.

The code distance d_{min} is one of the major parametres of error-correcting codes. Let consider a pairs of the allowed code words from a table. 3.1. It is possible to establish, that for this code a minimal distance is d_{min} =2. Such distance allows to *detect a single errors* in the channel. If the transmitted code word is $\boldsymbol{b}$=1 1 0, and the error in the channel is characterised by a word (*an error vector)* ***e***=0 1 0 the received word $\hat{b}$ with an error on the channel exit is defined by module-2 addition:

$$\boldsymbol{b}=1\ 1\ 0,$$
$$\boldsymbol{e}=0\ 1\ 0,$$
$$\hat{\boldsymbol{b}}=(\boldsymbol{a}\oplus\boldsymbol{e})=1\ 0\ 0.$$

From this it is visible, that the symbol «1» in the error vector ***e*** *changes* the corresponding symbol in transmitted word ***b*** to an opposite symbol.

For the characteristic of quantity of the channel errors enter concept the *brevity* of an errors.

The *brevity of an errors q is* a quantity of the channel errors within a codeword.

For example, for words from table 3.1 the error vector variants with brevity *q=1* are ***e***=100, 010, 001. And the *double errors* are:110, 011, 101.

The code ability to detect and to correct of the errors depends from the code distance d_{min} .

Error detection is the fixing by decoding of the error presence of certain brevity in the received word $\hat{\boldsymbol{b}}$.

Error-correction is the *detection* by decoding of an errors in the certain symbols of the received words and their subsequent *correction.*

According to these definitions error-correcting codes are subdivided into following *classes:*

1. The *error-detecting codes* which detect of the channel errors
2. The *eror-correcting codes* which correct of the channel errors and named in the literature as *codes with direct correction of errors* (i.e. with errors *correction by a code methods*).

The relation between code distance d_{min} and error-correcting ability of a code we will establish on an example of a code with even number of units (see table. 3.1). It is convenient to use a geometrical representation of code words represented on figure 3.1. Let's represent a code words by set from three symbols (*x, y, z*), and values of these symbols will choose from the binary alphabet {0,1}. It is possible to represent all possible code words by the points in the Cartesian system with coordinates (*x, y, z*). Thus words will form tops of a three-dimensional cube. On figure 3.1 these tops are marked as follows:

– By the sign "•" notes the *allowed* code words,

– By the sign "× " notes the *forbidden* code words.

It is visible, that the code structure is that *between the allowed code words are forbidden words*. They form *a «protective interval»*. Therefore the action of any single error translates any from the allowed word to the nearest forbidden words. This property leads to *the decoding rule* of a code with even number of units and detection of any single errors.

> *Reception* from the channel output *of the forbidden* code words allows to assert that in the channel there *was a single error*.

It is easy to be convinced that this code *does not allow to detect double errors (*because «*the protective interval*» is *nonsufficient)*. By an induction it is possible to prove, that *any binary code* with *even number of units allows* to detect any errors if their brevity is odd , and *does not detect* any errors if their brevity is *even*. The concept of «*a protective interval»* is easily applicable for a study of the relation between code distance and code ability to correct of an errors. If the minimum distance between the allowed code words (code distance) is d_{min}, that as is shown from figure 3.2 the protective interval contains *(*d_{min}*–*1*)* of the *forbidden words* and for "transfer" of each allowed word to the nearest allowed word it is necessary by errors to make (d_{min}–1) "*steps*". Clearly, that all errors with *brevity* q=1,2,3 … *(*d_{min}-1*) can be detect.*

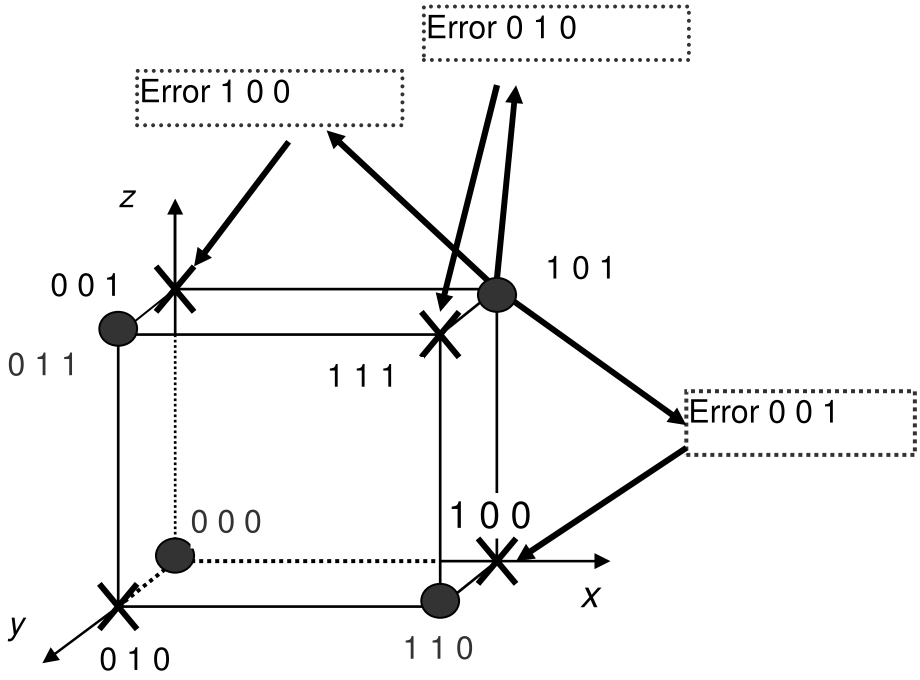


Figure 3.1– To an illustration of a code correction ability

From here follows, that:

> If the code distance of a binary code is d_{min} the *code ability to detect* and of the errors with brevity q_{det} is defined:

$$q_{\det} \le d_{\min} - 1$$

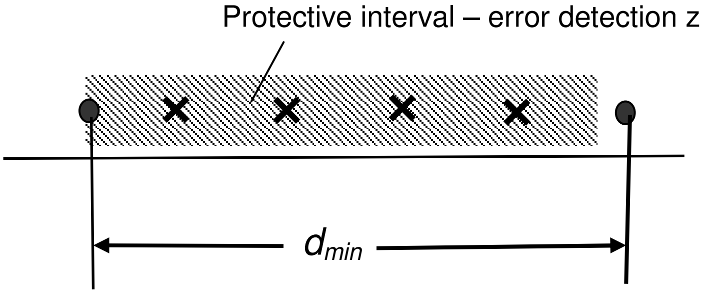


Figure 3.2 – To an illustration of a error-detecting ability

Let's take advantage of similar representation for estimations of ability to correct of the errors. On the figure 3.3 the layout of the allowed code words $\boldsymbol{b}_{\text{allow.1}}$ and $\boldsymbol{b}_{\text{allow.2}}$ is shown. Between them are allocated (d_{min}–1) the forbidden words. Let's divide all set of the words on two *allowed subset* as is shown in a figure 3.3. If, for example, the received word $\hat{b}$ is allocated into «*allowed decoding subset of a word* $\boldsymbol{b}_{allow.1}$» that during the decoding becomes decision about transmitting of the word $\boldsymbol{b}_{allow.1}$, i.e. thereby the transition errors of the word $\boldsymbol{b}_{\text{allow.1}}$ to the nearest forbidden words are corrected. It is similarly possible to explain error-correcting process by the transmission of the word $\boldsymbol{b}_{\text{allow.2}}$. It is visible, that the distance of each allowed subset is (d_{min}–1)/2 (by d_{min} is odd). It defines error-correcting ability of a code. For even values d_{min} the distance of each allowed subset is [(d_{min}/2)– 1], that also defines error-correcting ability of a code.

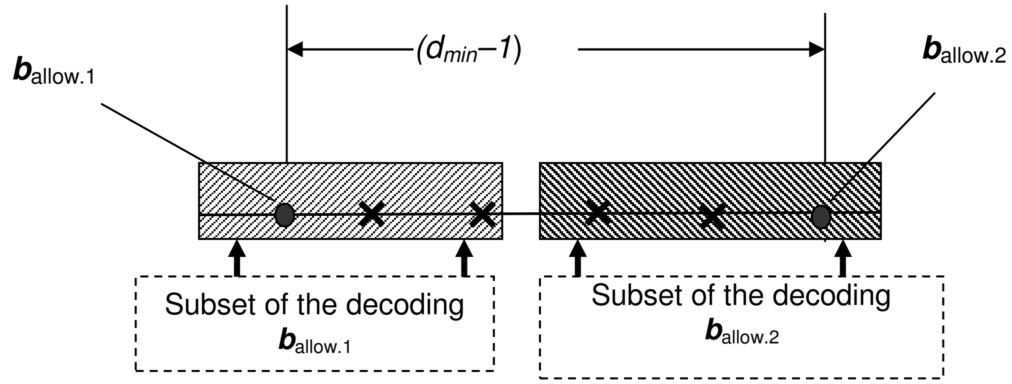

(dmin–1)
ballow.1
ballow.2
Subset of the decoding ballow.1
Subset of the decoding ballow.2

Figure3.3 – To an illustration of a error-correcting ability

Thus:

If the code distance of a binary error-correcting code is d_{min} *the code ability to correct* of the errors is defined by the expressions:

$$q_{corr.} \le \frac{d_{min} - 1}{2} \qquad (d_{min} – \text{odd});$$

$$q_{corr,} \le \frac{d_{min}}{2} - 1 \qquad (d_{min} – \text{even})$$

Lecture 4

Algebraic description of the block codes

Plan

4.1. Algebraic description of the block codes [2, Section 3.2]

Instruction. Study the elements of the general algebra from the manual [3, Section 2]

4.1. Algebraic description of the block codes

For the description of the linear block codes use a mathematical apparatus of the general algebra. By the block coding form code words $\boldsymbol{b} = (b_1, b_2..., b_n)$. Symbols of binary codes choose from the *Galua Field GF* (2). The set of words forms n-dimensional vector space over *Field GF* (2). For elements of this space (*vectors*) the addition and multiplication operations and operation of multiplication of a vector and also a scalar product of a vectors are defined. Some vectors subset of the space $\boldsymbol{B_n}$ which satisfy to the vector space axioms organises subspace $\boldsymbol{A_k}$.

The binary block code with block length n and 2^k allowed code words is called as the *linear (n, k) code* if its code words form k-dimensional vector subspace $\boldsymbol{A_k}$ of n-dimensional space $\boldsymbol{B_n}$.. Subspace $\boldsymbol{A_k}$ is generated by *the basis* from k linearly independent vectors, which organise the lines *of a generator matrix of the (n, k) code*:

$$\boldsymbol{G}=\begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,n} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k,0} & g_{k,1} & g_{k,2} & \cdots & g_{k,n} \end{bmatrix}. \qquad (4.1)$$

It is possible to present code words in the *systematic form,* forming separately *informational part* from k numerals and a *check part* from $r = (n– k)$ additional numerals.

The generator matrix of a systematic code looks like:

$$\boldsymbol{G}_{syst} = \left|I_k\, P\right| = \left|\underbrace{\begin{matrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{matrix}}_{\boldsymbol{I_k}}\ \underbrace{\begin{matrix}g_{1,n-r} & \cdots & g_{1,n}\\ g_{2,n-r} & \cdots & g_{2,n}\\ \vdots & \cdots & \vdots\\ g_{k,n-r} & \cdots & g_{k,n}\end{matrix}}_{\boldsymbol{P}}\right|. \qquad (4.2)$$

Matrix $\boldsymbol{G}_s$ contains *identity matrix* $\boldsymbol{I_k}$ wich defines the *information part* of a code words and matrix ***P*** defines the *additional simbols.* Transition to the systematic form is made by a *linear combination of rows* from the matrix (4.1). Such transition is illustrated by a following example.

Example 4.1. Matrix transformation of the nonsystematic code.

The nonsystematic block code (7,4) is set by the generator matrix:

$$\boldsymbol{G}_{\text{nonsist}} = \begin{vmatrix}1011010\\0100101\\0010011\\0001111\end{vmatrix} \qquad (4.3)$$

Using a method of the linear combination of rows from a matrix (4.3) we will transform it to the systematic form (4.2). For forming of a systematic generator matrix a rows of an initial matrix (4.3) it is convenient to present in the form of a table. 4.1. in which rows $\boldsymbol{g}1_{ns}$, $\boldsymbol{g}2_{ns}$, $\boldsymbol{g}3_{ns}$ and $\boldsymbol{g}4_{ns}$ are shown.

Table 4.1– Rows of the nonsystematic generator matrix

$\boldsymbol{g}1_{ns}$	1	0	1	1	0	1	0
$\boldsymbol{g}2_{ns}$	0	1	0	0	1	0	1
$\boldsymbol{g}3_{ns}$	0	0	1	0	0	1	1
$\boldsymbol{g}4_{ns}$	0	0	0	1	1	1	1

Using Modulo-2 addition rules the elements of these rows by exhaustive search of rows in various combinations it is established that by the most suitable variants for the forming of a matrixes rows for the systematic code are the following:

$$\boldsymbol{g}1_{syst.}=(\boldsymbol{g}1_{ns}\oplus\boldsymbol{g}3_{ns}\oplus\boldsymbol{g}4_{ns}),\ \boldsymbol{g}2_{syst.}=(\boldsymbol{g}2_{ns}\oplus\boldsymbol{g}3_{ns}),\ \boldsymbol{g}3_{syst.}=\boldsymbol{g}3_{ns},\ \boldsymbol{g}4_{syst.}=\boldsymbol{g}4_{ns}\ .$$

The outcome of an evaluation of the matrixes rows of the systematic code is reduced in table 4.2.

Table 4.2– The matrixes rows of the systematic code

$\boldsymbol{g}1_{syst}$	1	0	0	0	1	1	0
$\boldsymbol{g}2_{syst}$	0	1	0	0	1	0	1
$\boldsymbol{g}3_{syst}$	0	0	1	0	0	1	1
$\boldsymbol{g}4_{syst}$	0	0	0	1	1	1	1

The matrix of the systematic code in the standard form low given:

$$\boldsymbol{G}_{syst} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix} \qquad (4.4$$

> *The Hamming weight* $\boldsymbol{w}_H$ of the binary code word is equal to an amount of units in a code word.

In the block codes theory the important role plays the concept *weight of a code word.*

Example 4.2. An evaluation of the *Hamming weights* of the code words.

We will define values of the Hamming weights for the code words by table. 4.3:

Table 4.3 – The Hamming weights of the code words

Bynary code words							Weight $w_{\mathbf{H}}(\boldsymbol{b}_{\mathrm{i}})$
$\boldsymbol{b}_1$	1	0	1	1	0	1	4
$\boldsymbol{b}_2$	0	1	0	0	0	1	2

The strukture of a generator matrix allows to define the minimum distance of the block codes. This position is illustrated by following exercise.

Exercise 4.1. Definition of the code by its generator matrix.

Generator matrixes of the error-correcting codes by (4.3) or (4.4) are set. Show how to define code distance of a codes by a known generator matrix.

Instruction. By elaborating of a method for the definition of a code distance it is necessary to consider, that the zero combination $\boldsymbol{b}_0$ = (00 … 0) also is allowed.

Decision. It is above noticed, that the allowed code words are defined by a linear combinations of the rows of a generator matrix. As the zero word $\boldsymbol{b}_0$ = (00 … 0) also is allowed, and rows of a generator matrix $\boldsymbol{g}1$, $\boldsymbol{g}2$, $\boldsymbol{g}3$, $\boldsymbol{g}4$ also are the allowed words then Hamming distances from these words to a zero word $\boldsymbol{b}_0$ it is defined their weights $d_H(\boldsymbol{g}i, \boldsymbol{b}_0) = w_H(\mathbf{g}i)$, $i = (1 \ldots k)$. Further it is necessary to find the minimum weight, i.e. the minimum distance. Such conclusion from here follows:

> *The code distance* as the value of the minimum distance between the allowed code words is defined *by the least weight of rows* of a generator matrix .

Example 4.3. Definition of the Hamming weights of the generator matrix rows of a systematic code. Define values of the rows weights of a generator matrix from the Example 4.1 (table 4.2). Outcomes of evaluations are reduced in table 4.4.

Table 4.4– The Hamming weights of the generator matrix rows
for the systematic code

The generating matrix rows								Weights $w_H(\boldsymbol{g}_i)$
$\boldsymbol{g}1_{syst}$	1	0	0	0	1	1	0	3
$\boldsymbol{g}2_{syst}$	0	1	0	0	1	0	1	3
$\boldsymbol{g}3_{syst}$	0	0	1	0	0	1	1	3
$\boldsymbol{g}4_{syst}$	0	0	0	1	1	1	1	4

The analysis of data makes definition of the minimum distance of the systematic code from table. 4.4 $d_{min\,(\mathrm{syst.})} = min\ \{w_H(\boldsymbol{g}_i)\} = 3$.

Exercise 4.2. Define by the same way the code distance of nonsystematic code from an Example 4.1 (table. 4.1).

Instruction. The statement about code distance of a block code from Exercise 4.1 is fair both for the systematic and for nonsystematic codes.

Decision. We will apply a technique from the Example 4.3. Outcomes of evaluations the weigths of rows are reduced in table. 4.5.

Table 4.5 – The Hamming weights of the generator matrix rows
for nonsystematic code

Generator matrix rows								Weights $w_H(\boldsymbol{g}_i)$
$\boldsymbol{g}1_{ns}$	1	0	1	1	0	1	0	4
$\boldsymbol{g}2_{ns}$	0	1	0	0	1	0	1	4

$\boldsymbol{g}3_{ns}$	0	0	1	0	0	1	1	3
$\boldsymbol{g}4_{ns}$	0	0	0	1	1	1	1	4

The analysis of these data makes definition of the minimum distance of nonsystematic code from таб.4.5 $d_{Hmin\ (\mathrm{ns})} = min\ \{w_{\mathrm{H}}\ (\boldsymbol{g}_{\mathrm{i}})\} = 3$. The received outcomes allow to state that systematic and nonsystematic codes from the Example 4.1 on the value of code distance *are equivalent.*
Thus:

> *The code distance of a block code is a least weight of nonzero rows from the code generator matrix.*

The above noted dependence between the minimum distance of block codes and weights of nonzero rows can be used for forming of a generator matrix of a block code with the beforehand set code distance. This is illustrated by outcomes of
an Examples 4.4 and 4.5.

Example 4.4. A generator matrix of a code with even number of units.

Let's form the generator matrix of the systematic (*n, k*) code which *detect a single errors* (q_{det}=1). Such code should have code distance $d_{min}= q_{det}+1=2$. Hence, the nonzero rows of a generator matrix of this code should have the minimum weight w_H=2. According to the standard form (4.4) each row of the systematic code matrix already contains a numeral 1 (defined by an submatrix $\boldsymbol{I}_{\text{к}}$), the weight should be increased the weight of every rows to 2 having added in last numerals every rows (as a part of submatrix $\boldsymbol{P}$) a numeral 1.

For an example the generator matrix such (7,4) codes with k=4 will look like

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{vmatrix}, \qquad (4.5)$$

and unit in submatrix $\boldsymbol{P}$ can be in any place of a line.

Exercise 4.3. Generator matrixes of the codes which can detect double errors.

Form generating matrix of the systematic code which can detect double errors.

Instruction. From the theory does not follow that such codes there can be only one. It is recommended to consider at first a principle construction of a matrix at least one code and then on this basis to give generalisation and to find matrixes of several more codes.

Decision. The code can detect errors with brevity q_{det}=2 should have the minimum distance d_{min}= q_{det}=2+1=3. Hence, rows of a generator matrix of such code should have the minimum weight w_H=3. From a general view of a generator matrix of a systematic code (4.2) follows what to get such weight it is possible by the choice of rows of the submatrix of additional symbols ***P***, and one of row of this submatrix should have the weight equal 2.

Following variants of submatrix ***P*** are possible:

$$\boldsymbol{P}_1=\begin{vmatrix}0&1&1\\1&1&0\\1&0&1\\1&1&1\end{vmatrix};\ \boldsymbol{P}_2=\begin{vmatrix}1&1&0\\0&1&1\\1&1&1\\1&0&1\end{vmatrix},\ \boldsymbol{P}_3=\begin{vmatrix}1&1&1\\1&0&1\\1&1&0\\0&1&1\end{vmatrix}, \qquad (4.6)$$

which differ permutation of rows. As the minimum of each weight rows of these matrixes is equal to 2, they can be used for forming of systematic codes with the minimum distance d_{min}=3. In particular, the generator matrix of one of such codes looks like:

$$\boldsymbol{G}=\begin{vmatrix}1&0&0&0&0&1&1\\0&1&0&0&1&1&0\\0&0&1&0&1&0&1\\0&0&0&1&1&1&1\end{vmatrix} \qquad (4.7)$$

Questions

4.1. Codes with generator matrixes (4.3) and (4.4) have the identical minimum distance. As it to explain ?

4.2. Whether error-correcting properties of a block code vary by permutation of columns of a generator matrix?

4.3. Whether is the only the one method of transformation from nonsystematic generator matrix to the systematic code matrix for forming which is considered in the Example 4.1?

Tasks

4.1. Offer a transformation method of a generator nonsystematic matrix (4.3) for deriving of the systematic matrix which is alternative to a method from Example 4.1.

4.2*. Following a method stated in Exercise 4.3, form generator matrix of the systematic code which correct triple errors. Is the discovered by you code the unique?

Lecture 5

Encoding and decoding of the block codes

Plan

5.1.Encoding and decoding of the block codes[2, Section 3.3]
5.2.Syndrome decoding of the block codes
5.3. Majority decoding of the block codes

5.1. Encoding and decoding of the block code

In the center of the block coding theory is the concept of a generator matrix (4.1) and (4.2).

If $\boldsymbol{a}$ = | a_0, a_1..., a_k | – row-matrix of a primary code the *coding* make by a rule: $\boldsymbol{b=a{\cdot}G}$.

The *coding rule* of a block code is defined by the product

$$\boldsymbol{b=aG,} \tag{5.1}$$

where $\boldsymbol{a}$ = | a_0, a_1..., a_k | – row - matrix of a primary code at the encoder input,
$\boldsymbol{b}$ = | b_0, b_1..., b_n | – row - matrix of a block code word at the encoder output,
$\boldsymbol{G}$ – a generator matrix of the linear *(n, k)* code.

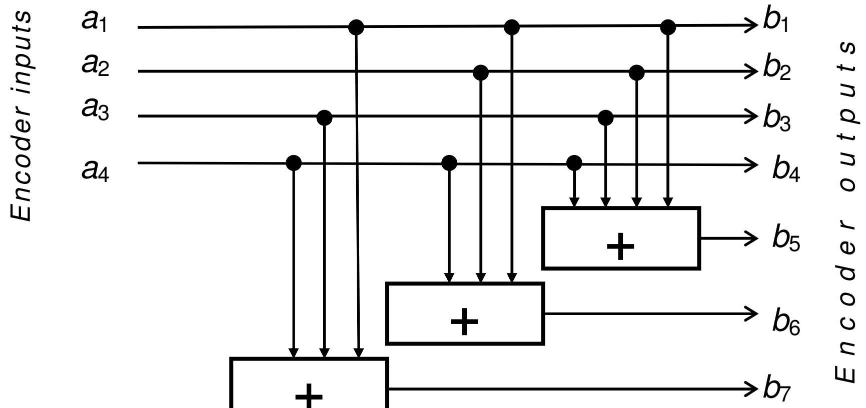
Example 5.1. The encoder of a code (7,4).

The structure of the encoder of a systematic code (7,4) is defined by a generator matrix (4.4) and a rule of coding (5.1).

If on an encoder input is the symbols row of a primary code $\boldsymbol{a} = (a_1, a_2, a_3, a_4)$ then symbols of the *allowed code word* on its output $\boldsymbol{b} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$ are defined by following equalities:

$$b_1=a_1,\ b_2=a_2,\ b_3=a_3,\ b_4=a_4,\ b_5=a_1\oplus a_2\oplus a_3\oplus a_4,\ b_6=a_1\oplus a_2\oplus a_4,\ \mathrm{b}_7=a_1\oplus a_3\oplus a_4. \tag{5.2}$$

On figure 5.1 the structure of the encoder of a systematic code (7,4) with equalities (5.2) is shown.



By the decoding of block codes the *check relations* establish with use of the *parity check matrix* ***H*** which space of rows *is orthogonal* to space of rows of a generator matrix, that is:

$$\boldsymbol{G}\cdot\boldsymbol{H}^{T}=\mathbf{0}. \qquad (5.3)$$

Here *T*– an index a transposition.

> The rows space of the parity check matrix ***H*** is *orthogonal* to the rows space of the generator matrix ***G****:*
>
> $$\boldsymbol{G}\cdot\boldsymbol{H}^{\mathrm{T}}=0.$$

If the generator matrix is set in the form (4.2) for performance of a orthogonality condition *the parity check matrix* should look like:

$$\boldsymbol{H} = |\boldsymbol{P}^{\mathrm{T}} : \boldsymbol{I}_{\mathrm{n-k}}|, \qquad (5.4)$$

where $\boldsymbol{P}^{\mathrm{T}}$– transposed submatrix ***P*** of generator matrix ***G***,
$\boldsymbol{I}_{n-k}$– an identity matrix a size [(*n – k)* × (*n–k)*].

Exercise 5.1. A parity check matrix of a systematic code (7,4).

The generator matrix of a systematic code (4.7) is set:

$$\boldsymbol{G}_{syst.}=\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}$$

According to a rule (5.4) form the parity check matrix of this code.
Solution. Sequentially we discover the submatrixes entering into the formula (5.4):

–The transposed submatrix a size [(n– k) ×k]:

$$\boldsymbol{P^T}= \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{vmatrix},$$

–The identity submatrix a size $[(n–k) \times (n–k)]$:

$$\boldsymbol{I}_{(n-k)}= \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

We unite submatrixes in the uniform parity check matrix of a code:

$$\boldsymbol{H}_{\text{syst.}}= \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}. \qquad (5.5)$$

From a condition of orthogonality of generator and parity check matrixes of a linear code (5.3) follows that each allowed word of a linear code generated by a rule $\boldsymbol{b = aG}$ also satisfies to an orthogonality condition:

$$\boldsymbol{b \cdot H^T = a \cdot G \cdot H^T=0}. \qquad (5.6)$$

By transmission through the channel code symbols are distorted. The received words look like $\hat{b} =\boldsymbol{b}\oplus\boldsymbol{e}$, where $\boldsymbol{b} = (b_0 b_1... b_n)$, and an error vector $\mathbf{e} = (e_0e_1... e_n)$.

By decoding calculate *a syndrome vector*

$$\mathbf{S} =\hat{\boldsymbol{b}} \cdot \mathbf{H}^{\mathrm{T}} =(s_0\, s_1...s_{n\text{-}k\text{-}1}). \qquad (5.7)$$

The syndrome depends only from a error vector:

$\mathbf{S} =\hat{\boldsymbol{b}} \cdot \mathbf{H}^{\mathrm{T}}=(\mathbf{b}\oplus\mathbf{e})\ \mathbf{H}^{\mathrm{T}}=\hat{\boldsymbol{b}} \cdot \mathbf{H}^{\mathrm{T}}\ \oplus\mathbf{e}\cdot\mathbf{H}^{\mathrm{T}}$. As the condition of orthogonality $\hat{\boldsymbol{b}} \cdot \mathbf{H}^{\mathrm{T}}=\mathbf{0}$ is satisfied, the *syndrome* is equal:

$$\boldsymbol{S = e \cdot H^T} \qquad (5.8)$$

From here the simple rule of an error detection follows:

> *If the syndrome* $\boldsymbol{S}$*=0* then an error vector **e**=*0,* i.e. *in the channel there were no errors* and the received word belongs to set of the allowed code words.
>
> *If S≠0* word $\hat{b}$ *contains errors.* It is possible on the syndrome symbols to define a *configuration of a error vector.*

This principle underlies *syndrome decoding.*

5.2. Syndrome decoding of the block codes

The principle of syndrome decoding we will consider on an example of a simple block code.

Example 5.3. The syndrome decoder of the systematic code (7,4).

According to a rule (5.8) for realisation of the syndrome decoder it is necessary to form *the transposed parity check matrix* of a code (7,4). The parity check matrix of this code looks like (5.5). Applying to it a rule of a transposition of matrixes it is received:

$$\boldsymbol{H}_{\text{syst.}}=\begin{vmatrix}0&1&1&1&1&0&0\\1&1&0&1&0&1&0\\1&0&1&1&0&0&1\end{vmatrix}. \qquad \boldsymbol{H}_{syst.}{}^{T}=\begin{vmatrix}0&1&1\\1&1&0\\1&0&1\\1&1&1\\1&0&0\\0&1&0\\0&0&1\end{vmatrix} \qquad (5.9)$$

It is convenient to note the syngle errors from a telecommunication channel so:

$$\boldsymbol{e}_1=(100\ldots0),\ \boldsymbol{e}_2=(010\ldots0),\ \boldsymbol{e}_3=(001..0),\ \ldots,\ \boldsymbol{e}_{\text{n}}=(000..1). \qquad (5.10)$$

In such form the error vector $\boldsymbol{e}_{\text{i}}$ represents a symbol set from n elements in which on a place with number i the symbol of an error 1 (at the left) is arranged and on remaining places zero symbols are arranged. Error vectors can be presented in the form of an identity matrix:

$$\boldsymbol{E}=\begin{vmatrix}\boldsymbol{e}_1\\\boldsymbol{e}_2\\\cdot\\\cdot\\\boldsymbol{e}_{\text{i}}\\\cdot\\\boldsymbol{e}_{\text{n}}\end{vmatrix}=\boldsymbol{I}_{\text{n}}=\begin{vmatrix}1&0&0&0&0&0&0\\0&1&0&0&0&0&0\\0&0&1&0&0&0&0\\0&0&0&1&0&0&0\\0&0&0&0&1&0&0\\0&0&0&0&0&1&0\\0&0&0&0&0&0&1\end{vmatrix}, \qquad (5.11)$$

which each row is the single error vector. Using properties of identity matrixes, it is easy to show, that the matrix of syndromes coincides with the transposed parity check matrix of this code (5.9):

$$\boldsymbol{S}=\boldsymbol{E}\cdot\boldsymbol{H}^T=\boldsymbol{I}_{\text{n}}\cdot\boldsymbol{H}^T=\boldsymbol{H}^T \qquad (5.12)$$

By the syndrom decoding of a block code *the matrix of syndromes S coincides with* the transposed parity check matrix of a code $\boldsymbol{H}^T$.

It is the foundation for tabling of syndromes**.** The more low reduced table 5.1 of syndromes for a code (7,4) is made according to rows of the transposed parity check matrix (5.9) . In the table to each vector of an error there corresponds the vector of the syndrome specifying a location of an error symbol in the received code word.

Table 5.1 – The syndromes for decoding of a code (7,4)

Syndromes	011	110	101	111	100	010	001
Error	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$	$\boldsymbol{e}_5$	$\boldsymbol{e}_6$	$\boldsymbol{e}_7$

It allows to formulate of syndrom decoding algorithm.

The syndrom decoding algorithm of block codes is the following:

1.Forming of the transposed parity check matrix of a code $\boldsymbol{H}^T$.

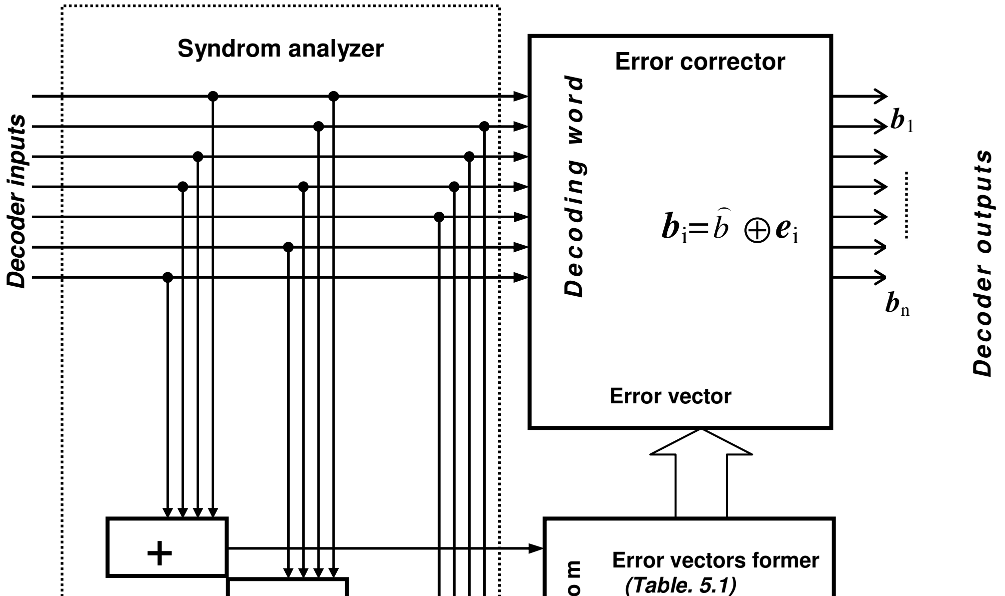
2.Tabling of syndromes for decoding of a(n, k) code.

3.An evaluation of syndromes (as tab. 5.1) on structure of the transposed parity check matrix of code $\mathbf{H}^T$ and a vector of error symbols of a decoded codeword by a rule (5.12).

4.Forming of a vector of an error $\boldsymbol{e}_{\mathrm{i}}$ on the basis of the syndromes table.

5. Error correction in the received code combination by a rule: $\boldsymbol{b}_i=\widehat{\boldsymbol{b}}\oplus\boldsymbol{e}$

The structure of the syndrom decoder of the code (7,4) realising this algorithm is reduced on figure 5.2. According to rule (5.12) received channel symbols move to modulo 2 adders.The connections with lines of channel symbols are available there where in rows of the transposed parity check matrix the symbol 1 is arranged.



In the scheme of the syndrom analyzer with according to given tab. 5.1 there is a transformation of syndrom vectors $\boldsymbol{S} = (s_0 s_1 \ldots\ s_{n\text{-}k\text{-}1})$ in the corresponding error vectors $\boldsymbol{e}$ which then move to the error corrector.

5.3. Majority decoding of the block codes

Some block codes suppose realisation of simple *majority algorithm w*hich is based on a possibility to express each information code symbol of a word by several ways through other received simbols. Let’s consider a systematic code (7,3):

$$\boldsymbol{G}=\begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{vmatrix}. \qquad (5.13)$$

To this matrix correspond transposed parity check matrix:

$$\boldsymbol{H}^T=\begin{vmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \qquad (5.14)$$

Let's designate the code combination received from the channel as

$$\boldsymbol{b}=(b_1,b_2,b_3,b_4,b_5,b_6,b_7).$$

As a considered code – systematic, first three symbols (b_1, b_2, b_3) are *information* symbols. Using *structural properties* of this code, it is possible to form during decoding process both *trivial* and *compound* estimations *of information symbols* which are presented to table 5.2. On the basis of columns of a parity check matrix (5.15) we will write down *verifying parities*:

$$b_1\oplus b_3\oplus b_4=0,\ b_1\oplus b_2\oplus b_3\oplus b_5=0,\ b_1\oplus b_2\oplus b_6=0,\ b_2\oplus b_3\oplus b_7=0, \qquad (5.15)$$

which allow to form *compound estimations*. For example, on the basis of the first equality from (5.15) follows *the compound estimation* of the first information symbol $b_1=b_3\oplus b_4$. *The trivial estimation* of this symbol also is, actually, this symbol $b_1=b_1$, as a code is systematic. Expressions for other information symbols are made similarly. They are presented in the table5.2.

Table 5.2– Majority decoding of the block code

Estimations of an information symbols		
Symbol b_1	Symbol b_2	Symbol b_3
Trivial		
$b_1=b_1$	$b_2=b_2$	$b_3=b_3$
Compound		
$b_1=b_3\oplus b_4$ $b_{1=}b_5+\oplus b_7$ $b_1=b_2\oplus b_6$	$b_2=b_4\oplus b_5$ $b_2=b_6\oplus b_1$ $b_2=b_3\oplus b_7$	$b_3=b_5\oplus b_6$ $b_3=b_7\oplus b_2$ $b_3=b_4\oplus b_1$

After formation of estimations they move on *a majority element* in which the decision on each information symbol is taken out «*on the majority of voices*».

For example, if estimations of an information symbol b_1 look like:

$$b_1=b_1=1,\ b_1=b_3\oplus b_4=1,\ b_{1=}\oplus b_5+\oplus b_7=1,\ b_1=b_2\oplus b_6=0,$$

in which the quantity of estimations «1» exceeds quantity of estimations «0» the majority element passes the decision «*on the majority*»: $b_1=1$. The compound estimations enumerated in tab. 5.2 are called *as orthogonal* estimations as *incoincident numerals* enter into them. The number of orthogonal estimations N and a breavity of errors $q_{\text{corr.}}$, corrected at majority decoding are in the ratio:

$$q_{\text{corr.}}\leq(N-1)/2. \qquad (5.16)$$

The code with a generator matrix (5.13) allows to form N=3 orthogonal estimations and, accordingly, *to correct unitary errors* in information symbols *by considerable simplification of decoding algorithm.* It is necessary to notice, that rules of formation of estimations can have *cyclic properties* that simplifies decoding procedure.

Example 5.4. Structure of the majority decoder for the code (7,3).

Let's generate structure of the majority decoder of a code (7,3) on the basis of estimations system from table. 5.2. It is easy to see, that *checks have cyclic properties.*

For example, indexes in compound estimations $b_1=b_3\oplus b_4$, $b_2=b_4\oplus b_5$ and $b_3=b_5\oplus b_6$ change on 1 towards increase. Taking into account it the structure of the decoder of the code (7,3) realising majority decoding algorithm looks like, shown on figure. 5.3. The decoder consists of the shift register, the switchboard on the input, operated from system for block synchronisation, schemes of estimation formation and a majority element. The decoder works as follows. At the beginning the switchboard on an input is established in position «1» and the decoded code word

$$\boldsymbol{b} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$$

is entered in the shift register cells. Thus on inputs of a majority element the compound estimations defined by tab. 5.2 operate both trivial and compound estimations. The decision about a transmitted information symbol b_1 is read out from an exit of a majority element. Then the switchboard is established in position «2» and there is on one symbol shift of the word. On this step, owing to cyclic properties of estimations the second information symbol are formed and the decision on an information symbol b_2 is read out from an exit of a majority element. Further process repeats up to reception on a output symbol b_3 etc.

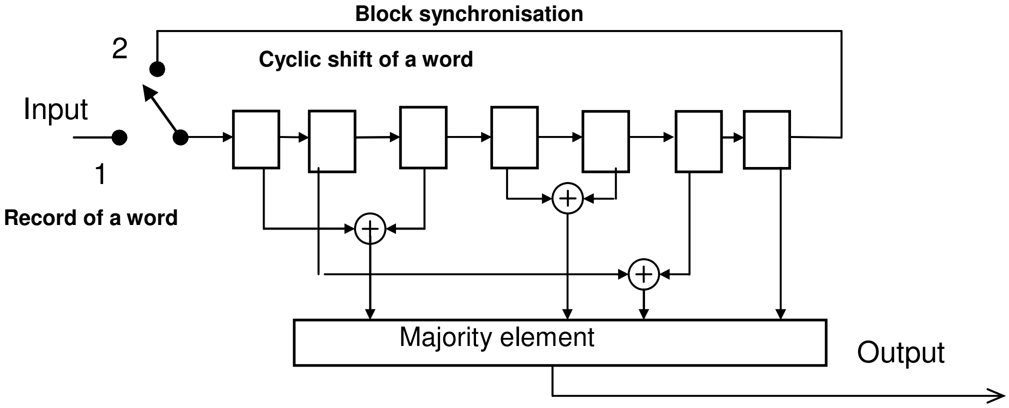


Figure 5.3 – Structure of the majority decoder of a code (7,3)

Questions

5.1. What kind will be have a matrix of double errors. How it will change in comparison with a matrix of single errors (5.11)?

5.2.How parametres of binary syndrome representation (table. 5.11 see) are connected with the general number of possible configurations variants which detected and corrected errors by syndrom decoding?

5.3.How the syndrome format will change if to apply a method of syndrome decoding to decoding double errors?

5.4.Result the generalised block diagramme of the syndrom decoder of a block code (n, k). What function is carried out by the syndrome analyzer?

Tasks

5.1***.**By the principles stated in the Example 5.1 represent structure a systematic block code intended for detection of double errors with the generating matrix (4.6).

5.2.The generator matrix of a code (7,4) is set:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Define the allowed code word of this code ***b*** if the word of a primary code on a coder input ***a*** = (1110) is set.

5.3.Define code distance of a code (7,4) with a generator matrix from the Task 5.2.

5.4.Represent a encoder structure of a code (7,4) with the same generator matrix.

Lecture 6
Boundaries of a parametres of the block codes

Plan

6.1.Boundaries of a parametres of the block codes[2, Section 3.3]

6.2. Hamming upper bound[2, Section 3.3]

6.3.Varshamov-Gilbert lower bound[2, Section 3.3]

6.4. Complexity of encoding and decoding algorithms[2, Section 3.3]

6.1. Boundaries of a parametres of the block codes

One of problems of the coding theory is the search of codes which at the given block length *n* and rate R_{code} provides a maximum of code distance d_{min}. Limits of these parametres are defined *by the code boundaries* which consideration is resulted more low.

6.2. Hamming upper bound

The conclusion of the upper bound is based on reasons *of spherical packing* (*bound of spherical packing*). At the given minimum distance between the allowed code word d_{min}. *the greatest rate* can be reached, if the spheres surrounding each word will be most densely packed.

The volume of each sphere is equal $\sum_{i=0}^{d_{\min}-1} C_n^i$, the number of spheres (number of code words) is equal 2^k. For the *best code* the total quantity of spheres and number of all possible words (2^n) should coincide. Equality is reached for *densely packed* (*perfect)* codes. The area of each code word represents sphere with radius ($d_{\min}$– 1)/2, and these areas of such codes not being crossed densely fill with themselves all *n*-dimensional space of code words. The inequality from here follows:

$$2^k \sum_{i=0}^{d_{\min}-1} C_n^i \le 2^n .$$

After simple transformations it is possible to receive obvious expression for rate of a perfect code:

$$1-R_{code} \ge \frac{1}{n}\log_2 \sum_{i=0}^{d_{\min}-1} C_n^i \; . \qquad (6.1)$$

The dependence of *Hamming upper bound* is shown on figure 6.1 *(*curve «Hamming upper bound»*). Hamming bound* is fair both for linear, and for nonlinear codes.

6.3.Varshamov-Gilbert lower bound

For block codes it is possible to get the *Varshamov-Gilbert lower bound* which defines the *possibility of codes existence* with both parametres R_{code} and d_{min}. The asymptotic form (for long codes) of this bound looks like:

$$R_{\text{code}} \ge 1-H(d_{min}/n), \qquad (6.2)$$

where *H* (*x)*– binary entropy. *The* dependence of *Varshamov-Gilbert lower bound* for binary codes is shown on figure 6.1 (Curve «Varshamov-Gilbert lower bound»). The *bound guarantees existence of the codes* which performances correspond to the points arranged at least on a curve (or above it).

||| *Search of the codes* ensuring the given minimum distance d_{min} and high

enough rate R_{code} at n→∞, ensuring at the same time a possibility of algorithms decoding realisation with low complexity *is one of the important problems of the theory of coding.*

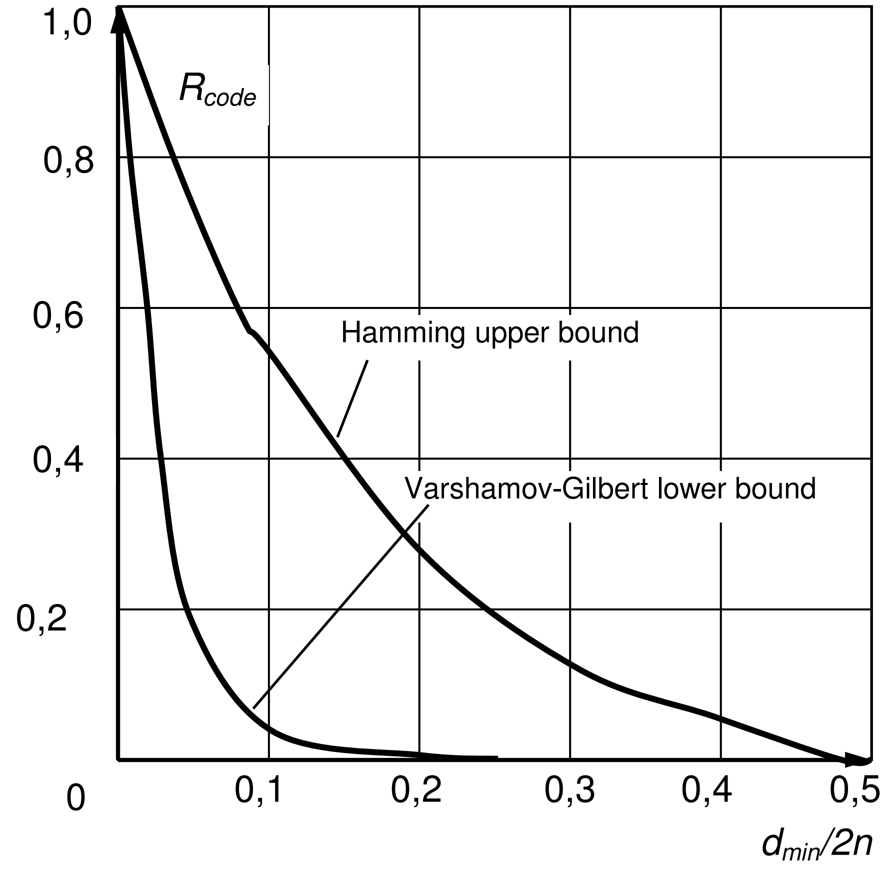


Figure 6.1– Code boundaries of block codes

6.4. Complexity of encoding and decoding algorithms

The using of correcting ability of a code depends on decoding algorithm. *By full decoding* use all possibilities to correct the errors following from properties of a code. According to Shannon *fundamental theorem* the error-correcting codes used for correction of channel errors should get out *long enough***.** However with growth of a code word length *n increases complexity of realisation of procedures of encoding and decoding* that causes *difficulty of practical realisation of codecs.*

In the coding theory of along with estimations of error-correcting ability of codes can *estimate complexity of realisation of encoding/decoding procedures* which can be realised by *software* or *hardware.* Thus as argument of complexity function the length of a codeword *n* should act.

Encoding complexity of a block codes $C_{encod.}$ with use of a generator matrix a (*n, k*) code with a size $nk{=}n^2\ (1-\ R_{\text{code}}\ )$ usually estimate in the value wich is proportional to number of elements of the generator matrix

$$C_{coding}=nk=n^2(1-R_{\text{код}}) \qquad (6.3)$$

The decoding algorithms appear more difficult. Among them it is considered to be the most difficult algorithm *full- search algorithm* according to which the decoder by the full searching *compares* the received code word with the set of all possible words and the decision on that transmited from the allowed word which appears on *the minimum distance* from the received word (*decoding by a distance minimum)* passes. It is considered to be *complexity of algorithm of full-search decoding* proportional to quantity of all possible code words *to volume of full search:*

$$C_{\text{decod.}}=m^n \qquad (6.4)$$

It is said that complexity of full-search decoding increases «*as an exponent* » with growth of length of a code. Clearly, that *full-search decoding algorithms are practically difficult for realising for long codes.*

Questions

6.1. What is practical significance of use of Hamming upper bound and Varshamov-Gilbert lower bound for an estimation of performances of block error- correcting codes?

6.2. To what bound (upper or lower) it is necessary to aspire by elaborating of new block codes?

Lecture 7

Important classes of block error-correcting codes

Plan

7.1. Hamming codes [1, Section 10.2]

7.2. Cyclic codes [1, Section 10.3]

Instruction. In lecture materials it is widely used mathematic algebras of polynomials. It is necessary to study preliminary section 5 «*Algebra of polynomials*» from the manual [3].

7.1. Hamming codes

The big number of codes, various on structure, construction principles and correcting ability is known. In this lecture the classes of effective block codes with simple decoding algorithms are considered.

Hamming codes(*by R. Hamming*)– Systematic block codes with parametres:

– Length of the code word $n=2^r-1$

– Quantity of information symbols $k=2^r-r-1$ (7.1)

– Number of additional symbols $r=n-k$,

– Minimum distance $d_{min}=3$.

$r=2, 3,4.$

Hamming codes – *perfect* codes which *correct single errors.*

By the parametre choice r=2, 3,4 according to formulas (7.1) it is possible to set all known binary Hamming codes. For example, at r=3 the parametres of a code (7,4) will be the following:

– Length of the code word n=3,

– Quantity of information symbols k=4;

– Minimum distance d_{min}=3;

– Code rate $R_{\text{code}}=(2^r-r-1)/(2^r-1)$=4/7.

Generator and parity check matrixes of this code have been considered earlier, in Section 4.1 (formulas (4.4) and (4.7)). As it has been noted earlier, this code allows to *detect double errors also.* Structures of the encoder and the syndrome decoder of a Hamming code have been considered earlier in Section 5.1 (figures 5.1, 5.2). According to the formula (3.5) transposed parity check matrix of this code looks like:

$$\boldsymbol{H}^{\mathrm{T}}=\begin{vmatrix}1&1&0\\1&0&1\\0&1&1\\1&1&1\\1&0&0\\0&1&0\\0&0&1\end{vmatrix}. \qquad (7.2)$$

7.2. Cyclic codes

The considerable part of block codes belongs to the class of cyclic codes. It defines a simplification of both encoding and decoding procedures on the basis of a cyclical properties of code words.If $\boldsymbol{b}$ = (b_0, b_1... b_{n}) – the allowed code word of the cyclic code so its cyclical shift on arbitrary number of symbols also is the allowed code word. For example, a word $\boldsymbol{b}^{(1)}$= (b_{n}, b_0, b_1..., $b_{\text{n-1}}$) corresponds to cyclical shift of a word $\boldsymbol{b}$ = (b_0, b_1... $b_{\text{n-1}}$, b_{n}) on one symbol to the right. Thus according to *a rule of cyclical permutation* combination symbols $\boldsymbol{b}$ are displaced on one numeral to the right, and the right numeral b_n takes a place of a left numeral b$_0$ Properties of the of the cyclic code are convenient for studying, representing code words in the form of polynomials on degrees of a formal variable x which factors are symbol numerals in a code word $\boldsymbol{b}$ (x) $=b_0+b_1x+b_2x^2$ +... $+b_n$ x^n. Mathematical operations (addition, multiplication and division of polynomials) make *by* rules of algebra of the polynomials stated in Section 5 of manual [3]. If addition and multiplication of polynomials is made by the modulo of a polynomial (x^n– 1) so all possible polynomials of degree (n–1) and less organise *an algebraic ring of polynomials* $\boldsymbol{R}_n$ with the properties stated in the manual [3].

For construction of a cyclic code in a ring $\boldsymbol{R}_n$ choose a subset of polynomials an *ideal* ***I.*** The polynomial of the minimum degree $\boldsymbol{g}$ *(x)* in this subset is called as a *generator polynomial* of the cyclic code . As generator polynomials of the cyclic code choose the *prime polynomials* .

In algebra of polynomials of the whole degree *prime polynomials* play the same role what *prime numbers* play int he algebra of integers. The detailed table of generating polynomials of cyclical codes is reduced in Attachment A.1. Generator polynomials of short cyclic codes are given in table. 7.1.

Table 7.1– Generator polynomials of short cyclic codes

Maximum degree of a generator polynomial	Generator polynomial $\boldsymbol{g}(x)$		
3	x^3+x^2+1	x^3+x+1	
4	x^4+x+1	x^4+x^3+1	
5	x^5+x^2+1	x^5+x^3+1	$x^5+x^4+x^2+1$
6	x^6+x+1	x^6+x^5+1	$x^6+x^5+x^3+x^2+1$

All polynomials of the ideal ***I*** corresponding to the allowed code words of the cyclic codes, are divided on the generator polynomial $\boldsymbol{g}(x)$ without remainder that allows to formulate a following en*coding rule*:

The encoding rule of nonsystematic cyclic code looks like:

$$\boldsymbol{b}(x)=\boldsymbol{a}(x)\boldsymbol{g}(x) \tag{7.3}$$

In practice often use systematic cyclic codes.

The *coding rule of systematic cyclic code* (*n, k*) looks like:

$$\boldsymbol{b}(x)=\boldsymbol{a}(x)x^{n-k}+\boldsymbol{r}(x), \tag{7.4}$$

where *r* (**x**)– remainder of division $\boldsymbol{a}$ (x)$x^{\,n-k}$ on $\boldsymbol{g}$ (*x*).

The coding rule (7.4) can be realised by such *coding algorithm* for a systematic cyclic code*:*

1.To the word of a primary code $\boldsymbol{a}$ *an it finish on the right* (*n*– *k*) zeros are added. It is equivalent to polynomial multiplication $\boldsymbol{a}$ on x^{n-k}.

2.Product $\boldsymbol{a}$ (*x*) x^{n-k} divides on the generator polynomial $\boldsymbol{g}$ (*x*). As a result of division remainder $\boldsymbol{r}$ (*x*) is defined.

3.The calculated remainder summarised with the displaced combination $\boldsymbol{a}$ x^{n-k}.

Therefore the *allowed code* word is formed:

$$\boldsymbol{b}(x)=\boldsymbol{a}(x)x^{n-k}+\boldsymbol{r}(x) \tag{7.5}$$

Example 7.2 Forming of a code word of the cyclic code (10,5**).**

For the given primary code word $\boldsymbol{a}$ = (10110) we will generate a code word of a cyclic code (10,5). Polynomial representation of the primary code word will be $\boldsymbol{a}$ $(x)=(x^4+x^2+x)$. At the given cyclic code are parametres n=10, k=5, r = (n-k) =5. From the table. 7.1 for example the generator polynomial $\boldsymbol{g}$ (x) = $(x^5+x^4+x^2+1)$ is chosen. Next we will fulfil mathematical operations according to algorithm (7.5):

$$\begin{array}{lr|l}
2)\boldsymbol{a}(x)\, x^{(n-k)}/\, \boldsymbol{g}(x) & =x^9+x^7+x^6 & x^5+x^4+x^2+1 \\
 & \oplus x^9+x^8+x^6+x^4 & \\
\hline
\oplus & x^8+x^7+x^4 & x^4+x^3+1 \\
 & \oplus x^8+x^7+x^5+x^3 & \\
\hline
 & x^5+x^4+x^3 & \\
\oplus & \oplus x^5+x^4+x^2+1 & \\
 & x^3+x^2+1=\boldsymbol{r}(x) &
\end{array}$$

1) $\boldsymbol{a}(x)x^{(n-k)}=(x^4+x^2+x)x^5=x^9+x^7+x^6$;

3) Polynomial of the allowed code word is

$$\boldsymbol{b}(x)=\boldsymbol{a}(x)x^{n-k}\oplus \boldsymbol{r}(x)=.\ x^9+x^7+x^6+x^3+x^2+1.$$

To polynomial $\boldsymbol{b}(x)=x^9+x^7+x^6+x^3+x^2+1$ there corresponds a word of binary symbols $\boldsymbol{b}$ = (1011001101) in which first four symbols are informational and remaining –
– additional.

Property of divisibility of the allowed code words of cyclic codes on the generator polynomial is widely used for *detection of errors* in telecommunication systems.

Property of divisibility of the cyclic code allowed code words on generating polynomial is widely used for *detection of errors* in telecommunication systems.

If $\hat{\boldsymbol{b}}(x)=\boldsymbol{b}(x)+\boldsymbol{e}(x)$– the received code word containing the errors polynomial $\boldsymbol{e}\ (x) =e_0+e_1x +... +e_nx^n$ as a result of division it is received:

$$\hat{\boldsymbol{b}}(x)/\boldsymbol{g}(x)=\boldsymbol{q}(x)+\boldsymbol{s}(x). \qquad (7.6)$$

Here: $\boldsymbol{q}$ *(x)* – an arbitrary polynomial ("whole"), $\boldsymbol{s}$ *(x)– the polynomial of a syndrome* equal *to* remainder of division $\hat{\boldsymbol{b}}(x)$ on $\boldsymbol{g}$ *(x).* It has degree not above (n– k– 1).

By absense of *errors a syndrome s (x) =0.*

On the syndrome form a it is possible *to establish* also *a location* of errors in the received code word and to use this information for *decoding with error-correction.*

Example 7.3. Syndrome decoding of words of a cyclic code (7,4).

The word of a binary primary code $\boldsymbol{a}$ = (1010) as subject to transmission via the channel with single errors is set. Let's choose the cyclic code ensuring errorless transmission this word in these conditions. From table A.1 we define, that the task can be solved by using of the cyclic code with a generatior polynomial $\boldsymbol{g}\ (x) = (x^3+x^2+1)$ and parametres n=7, k=4, $q_{corr.}$=1. We will show, how *the method of syndrome decoding* for correction of single errors is realised. Using algorithm of encoding (7.5), we will generate the allowed word $\boldsymbol{b}\ (x) = (x^6+x^4+1)$. We will suppose, that in the channel the single error $\boldsymbol{e}\ (x) = x^6$ operates. In this case the received word looks like$\hat{\boldsymbol{b}}(x)=\boldsymbol{b}(x)+\boldsymbol{e}(x)=x^6+x^4+1+x^6=x^4+1$. We use a rule for determination of a syndrome (7.6). By a syndrome decoding on the syndrome form it is possible to establish an error location (i.e. to fulfil *syndrome decoding).* For this purpose it is necessary to make *the table of syndromes* and of errors polynomials corresponding to them. For compiling of such table it is necessary to take advantage of the equality implying from (7.6) by $\boldsymbol{q}\ (x) = 0$:

$$\boldsymbol{s}(x)= \boldsymbol{e}(x)/\boldsymbol{g}(x) \qquad (7.7)$$

Outcomes of evaluations are presented to table 7.2 under this formula of syndrome polynomials $\boldsymbol{s}\ (x)$ for various polynomials of an errors. With a view of presentation a value of syndromes are presented in the form of binary words.

Table 7.2– Correspondence between syndromes and error polynomials

Error polynomial $\boldsymbol{e}(x)$	x^6	x^5	x^4	x^3	x^2	x	1
Syndrome $\boldsymbol{s}(x)$	x^2+x	$x+1$	x^2+x+1	x^2+1	x^4	x^2	1
Binary syndrome representation $\boldsymbol{s}$	110	011	111	101	100	010	001

Let the polynomial of the received from the channel word looks like $\hat{\boldsymbol{b}}(\mathrm{x})=x^4+1$. We will fulfil operation of division $\hat{\boldsymbol{b}}(x)/\boldsymbol{g}(x)$:

$$\begin{array}{l|l}
\quad x^4+1 & x^3+x^2+1 \\ \cline{2-2}
\oplus\, x^4+x^3+x & x+1 \\
\quad x^3+x+1 & \\
\oplus x^3+x^2+1 & \\ \hline
\quad x^2+x=\boldsymbol{s}(x) \text{ – Syndrome} &
\end{array}$$

From table. 7.2 it is discovered, that to such syndrome there corresponds an error polynomial $\boldsymbol{e}\ (x)=x^6$. Error correction consists in addition of the received code word with an error polynomial $\hat{\boldsymbol{b}}(x)+\boldsymbol{e}(x)=x^4+1+x^6=x^6+x^4+1$ that coincides with the transmitted allowed word $\boldsymbol{b}\ (x) = x^6+x^4+1$. To it there corresponds a binary word $\boldsymbol{b}$ = (1010101) in which first four symbols are errorless transmitted symbols of primary code $\hat{\boldsymbol{a}}$=(1010) (as the used code is systematic).

In practice finds application such codes with cyclic properties:

1. **Goley code** (23, 12) – perfect cyclic code with a generator polynomial $\boldsymbol{g}\ (x)=x^{11}+x^{10}+x^6+x^5+x^4+x^2+1$ and minimal distance d_{min}=7.

2. **Expanded Goley code** (24, 12) with minimal distance d_{min}=8 which receive by addition of the general parity checking.

3. **Bose, Ray-Chaudhuri, Hochuenghem codes** *(**BCH*** codes) which form extensive class of a cyclic codes.Bynary *BCH* codes have parametres: $n=2^m-1$, $(n-k)\leq mt$, $d_{min}=2t+1$, where *m* ($m\geq3$) and *t* – any positive integers.

Theoretical data on *BCH* codes are resulted in Section 10.4 of the textbook [1].

4. **Reed-Solomon codes**(***RS** codes*) – a subclass of nonbinary *BCH* codes with parametres: code symbols get out of field *GF* (*q*), $q=2^m$, *m* – whole; length of the word $N=(q-1)$, quantity of information symbols $K=(N-2q_{\text{corr.}})$, the minimum distance $D_{\min}=(2q_{\text{corr.}}+1)$. Possible also extention of a code to $N=q$ or to $N=(q+1)$.

The effective using of cyclic properties allowed words of cyclic codes allows to realise enough simple decoding algorithms . It is considered, that complexity of realisation of decoding algorithms of cyclic codes is described by sedate function $C_{decod.}=N^{\kappa}$, where the *k*– small number which size depends on concrete algorithm realisation. Examples of encoding/ decoding algorithms are more low resulted. Thus the mathematical apparatus of algebra of sedate polynomials and the description the discrete linear filters, presented in Sections 5 and 6 of the manual [3] is widely used.

Example 7.4. Encoder structure of the systenatic cyclic code.

Using algorithm (7.5) we will form the block diagramme of a cyclic encoder (15,11), with a generator polynomial $\boldsymbol{g}(x)=x^4+x+1$ which is chosen from table 7.1. The scheme of the encoder is resulted on figure 7.1. According to an algorithm (7.5) encoder works as follows. Originally switches S_1 and S_2 are in position 1. Eleven information symbols of a coded prime word $\boldsymbol{a}$ (*x*) are entered at the left into a chain of division into a polynomial $\boldsymbol{g}$ (*x*) $=x^4+x+1$. Simultaneously they through consistently connected delay elements arrive on a encoder exit, forming an information part of the allowed code word $\boldsymbol{a}$ (x) $x^{n\text{-}k}$. On first four steps in register cells the divider scheme on a generator polynomial the remainder of a division $\boldsymbol{r}$ (x) is formed. Then switches S_1 and S_2 are established in position 2, division process stops, and the remainder is read out from an exit of a divider and finished in a checking part of a final code word $\boldsymbol{b}(x)=\boldsymbol{a}(x)x^{n-k}+\boldsymbol{r}(x)$.

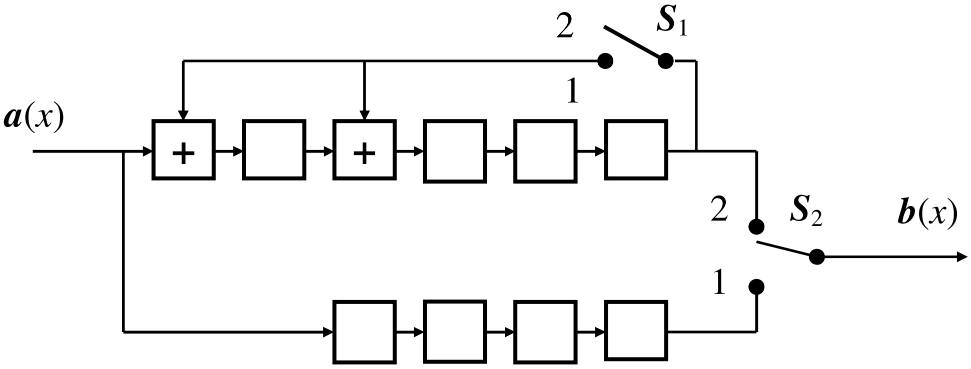


Figure 7.1 – Encoder of the syatematic cyclic code(15,11)

(+) –Adder on the module 2

Example 7.5. Encoder structure of the nonsystenatic cyclic code

Using a coding rule (7.3) for nonsystematic cyclic code we will form the coder block diagramme for a generator polynomial $\boldsymbol{g}(x)$ $=x^4+x+1$. The coding rule (7.3) provides multiplication of polynomials $\boldsymbol{a}$ (x) and $\boldsymbol{g}$ (x). Using structure of a multiplier for polynomials from section 6.1 of the manual [3] the encoder scheme we will present on figure 7.2.The importante element of schemes of coders and decoders for cyclic codes is the scheme of division polynomial on a polynomial for an evaluation of a division remainder by coding of a systematic code by algorithm (7.5) and also for a syndrome evaluation by syndrom decoding on the algorithm (7.6). The structure of such divider schemes is considered in Section 6.1 from the manual [3].

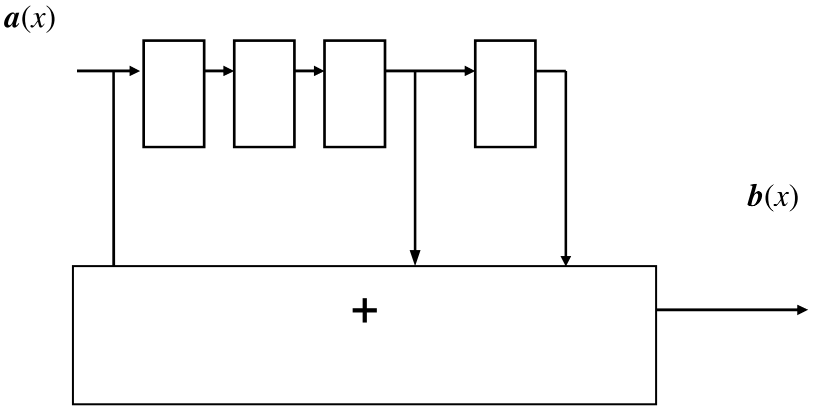


Figure7.2 – Encoder of the nonsystematic cyclic code

(+) –Adder on the module 2

Questions

7.1.What are the key parametres of Hamming codes?

7.2. What are the advantages of cyclic codes?

7.3. Whether it is possible to use Hamming codes and cyclical codes for correc tion of single errors? What will be parametres of these codes?

Tasks

7.1. The generatior matrix of a code (7,4) is set:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Define the allowed code word of this code ***b*** if the word of a simple code on a encoder input ***a*** = (1110) is set.

7.2. What aspect the parity check matrix of a code with a generator matrix from the Task 7.1 has?

7.3.The parity check matrix of a code (7,4) is set:

$$\mathrm{H} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Result a function chart of the decoder of this code.

7.4. Consider an example of formation of a allowed code word if a word of a simple code is ***a*** = (10010).

7.5*. By analogy to an example of Section 7.1 make the table of Hamming codes parametres for values *r* =2,3,4. As these codes have identical minimum distance, compare them on suitability for realisation in practical systems. Formulate the recommendation and a substantiation of application of the best (in your opinion) a code from this list.

7.6*. For Hamming code recommended in the previous Section, form generator and parity check matrixes.

7.7. By the rules stated in Exercise 4.1 define value of the minimum distance by a generatior matrix of code from the Task 7.5.

Lecture 8

Decoding noise immunityof the block codes

Plan

8.1. Decoding noise immunityof the block codes [1, Section 10.7]
8.2.Energy coding gain

8.1. Decoding noise immunityof the block codes

Let's define of an error probabilityby decoding of block codes in the binary symmetric channel. We will consider a code (*n, k)* with minimal distance d_{min}. In such channel an errors in sequentially transmitted code symbols (signals) occur independently with probability $P_{\text{err.}}$(decoding in the discrete channel without memory*).* Then the probability of that on length of the block *n* will occur a error brevity *j,* will be equal:

$$P_j = \mathrm{C}_n^j P_{\text{err.}}^j (1 - P_{\text{err.}})^{n-j.}.$$

Here C_n^j – number of combinations from *n* elements on *j.* If the code corrects all errors of brevity $q_{corr.}$=(d_{min}-1)/2 (d_{min}– odd) and less then the probability of reception on an decoder exit the word with not corrected errors will be equal:

$$P_{\text{err. word}} = \sum_{j=q_{\text{corr.}}+1}^{n} \mathrm{Pj}.$$

Hence, the probability of erroneous decoding of the block will satisfy to an inequality:

$$P_{\text{err.word}} \le \sum_{j=q_{\text{corr.}}+1}^{n} \mathrm{C}_n^j \mathrm{P}_{\text{err.}}^j (1 - \mathrm{P}_{\text{err.}})^{n-j}. \tag{8.1}$$

In this expression equality takes place, if the perfect code is used. Parities between parametres *n, k* and $q_{corr.}$ are defined by the concrete chosen code.

Expression (8.1) allows to define the upper estimation of error probability of an code words by decoding of the block codes in the binary symmetric channel without memory. For calculation of probability of an error in concrete information (or additional) symbols it is necessary to know used decoding algorithm and structure of an

error- correcting code(in particular, a set of distances from a transmitted code word to all allowed words). Such data for block codes are not published in a code tables and for calculations of probability of error decoding of code symbols (information or additional) use the approximated formula[1]:

$$p \approx \frac{1}{\mathrm{n}} \sum_{\mathrm{j=q_{corr.}+1}}^{\mathrm{n}} \mathrm{C_n^j P_{err.}^j (1-P_{err.})^{n-j}} \qquad (8.2)$$

For channels with coherent receiving of signals with binary phase modulation (PhM-2) the probability of an signal error reception is defined by the formula:

$$P_{\mathrm{err.}} = 2\mathrm{Q}(\sqrt{2}\mathrm{h}), \qquad (8.3)$$

where $\mathrm{h}^2 = \frac{\mathrm{E_s}}{\mathrm{N_0}}$ – the ratio of binary signal energy E_{s} to power spectral density of noise N_0 on a demodulator input;

$Q(z = \frac{1}{\sqrt{2\pi}} \int_{z}^{\infty} \exp(-\frac{t^2}{2})dt$ – special mathematic Q-function (probability integral) which tables contain the handbooks on probability theory and statistical calculations. For practical calculations it is convenient to use enough exact approximation:

$$Q(z) \approx 0.65\, exp[-0{,}44(z+0{,}75)^2]. \qquad (8.4)$$

The introduction of redundancy by using of error-correcting coding leads to expansion of a frequency band that occupied with a coded signal.If the frequency band in system without coding is F_0 (Hz) the using of a code with a rate

$R_{code} = \frac{\mathrm{k}}{\mathrm{n}}$ demands

expansions of a frequency band:

$$F_{code} = F_0 \frac{1}{\mathrm{R_{code}}} (\text{ Hz}). \qquad (8.5)$$

I.e. there is an expansion of a frequency band in $K_F = \frac{\mathrm{n}}{\mathrm{k}}$ time. For codes with low rate ($\frac{\boldsymbol{n}}{\boldsymbol{k}} > 1$) such expansion can appear appreciable. Therefore the problem of a code choice by designing of telecommunication system consists in search of *a compromise* between desirable degree of a noise immunity and expansion of a frequency band of the coded signal. Under formulas (8.2) and (8.3) taking into account expansion of a frequency band of coded signal according to the formula (8.4) following conclusions allow to draw on efficiency of application error-correcting coding :

1.With growth of a code word length *n* the error probability of an decoding *p* goes down;

2.Codes with the big redundancy (small code rate R_{code}) provide considerable decreasing of a decoding probability error;

3. By using of error-correcting codes in telecommunication systems *as a payment for noise immunity increasing* is *expansion of frequency band* of a transmitted signal, caused by the redundancy entered by coding on size:

$$K_F=\frac{n}{k} \tag{8.6}$$

8.2. Energy coding gain

For practice the question about expediency of application of error-correcting codes in telecommunications is important. This question decided taking into account the following. Introduction of redundancy by encoding changes not only expenses of a frequency band for transmitting of the coded signals, but also demands the account of a redundancy by energy calculations. Really, according to the formula (8.3) for probability of an error registration of channel signals (code symbols) is defined by their energy E_s which taking into account redundancy of a code appears a little bit less energy E_b spent for transfer of one information symbol (bit). It follows from equality $kE_b=nE_s$, i.e. $E_s=E_bR_{code}$. Therefore in all power calculations of systems with coding using, as a rule, the value of the ratio of signal energy spent for transmitting of one information binary symbol (bit) to noise power spectral density E_b/N_0. The probability of errorneous decoding of the block is defined by formulas (8.1) and (8.2) which in argument of function *Q (z)* include value E_s – the energy of a signal spent for transmitting through the channel of one binary signal (a code symbol). Really, according to the formula (8.3) probability of an error of registration of channel signals (code symbols) is defined by their energy E_s which taking into account redundancy of a code, appears a little bit less energy E_b spent for transmitting of one information symbol (bit). Then used in power calculations of systems with coding the relation of energy $E_b$to noise power spectral density N_0 can be designated as $h_b^2=\frac{E_b}{N_0}$.Taking into account relation of signal energy E_s and bit energy E_b(8.3) the value entering into the settlement formula will be $h^2=R_{code}h_b^2$. Then taking into account expenses of energy for transmitting of additional symbols of a redundancy code (8.3) it is possible to present the formula as follows:

$$P_{error}=2Q(\sqrt{2R_{code}}h_b), \tag{8.7}$$

and the bit error probability by expression

$$p\approx\frac{1}{n}\sum_{j=q_{corr}+1}^{n}C_n^jP_{error}^j(1-P_{error})^{n-j}, \tag{8.8}$$

in which the probability of an channel signal error is defined under the formula (8.3). If necessary to define probability of an error in the channel without coding it is enough to take advantage of the formula (8.3), having put R_{code}=1:

$$P_{\text{error}}=2Q(\sqrt{2}h_b). \qquad (8.9)$$

Exercise 8.1. Decoding noise immunity of a block code

Let's take advantage of the formula (8.9) for calculations of an error probability with optimum receiving of signals PhM-2 in the channel without coding. Results of calculations are resulted in table 8.1. Initial parametre for calculations is the relation a signal/noise on demodulator input $h_b^2=\frac{E_b}{N_0}$.The used in practice value $h_b^2(\text{dB})$define by formula $h_b^2(\text{dB})=10\lg h_b^2$. In table 8.1 settlement data by definition of error probability by optimum receiving of signals PhM-2 (formula (8.8)) including argument *z* of the function *Q (z).*

The dependence curve *p=f* (*h_b^2*(*dB*)) constructed on these data (PhM-2) is resulted on figure 8.1.

Table 8.1– Calculation of a receiving noise immunity of signals PhM-2

h_b^2, dB	*h_b*	*R_{code}*	*z*	*p*
1	1,122	1	1,587	0,12
2	1,259	1	1,178	$8\ 10^{-2}$
3	1,413	1	1,998	$4{,}7\ 10^{-2}$
4	1.585	1	2,241	$2{,}5\ 10^{-2}$
5	1,778	1	2,515	$1{,}2\ 10^{-2}$
6	1,995	1	2,822	$4{,}7\ 10^{-3}$
7	2,512	1	3,166	$1\ 10^{-3}$
8	2,818	1	3,986	$7\ 10^{-5}$
10	3,162	1	4,472	$8\ 10^{-6}$

Under formulas (8.7), (8.8) we will define of an bit error probability by decoding in the channel with PhM-2 words of a cyclic code average length (31,26) with parametres R_{code}=0,84, q_{corr}=1. The code is chosen from table A.1.

Table-8.2– Calculation of a decoding noise immunity of the cyclic code

Modulation method PM-2, Cyclic code (31,26)					
h_b^2, dB	*h_b*	*R_{code}*	*z*	C^2_{31}	*p*

1	1,122	0,84	1,454	$20\ 10^{6}$	$3\ 10^{-2}$
2	1,259	0,84	1,632	$20\ 10^{6}$	$3,1\ 10^{-2}$
3	1,413	0,84	1,831	$20\ 10^{6}$	$2\ 10^{-2}$
4	2,585	0,84	2,054	$20\ 10^{6}$	$1,0\ 10^{-2}$
5	1,778	0,84	3,305	$20\ 10^{6}$	$5\ 10^{-3}$
6	1,995	0,84	2,586	$20\ 10^{6}$	$4,2\ 10^{-3}$
7	2,239	0,84	2,902	$20\ 10^{6}$	$1,9\ 10^{-3}$
8	2,512	0,84	3,562	$20\ 10^{6}$	$2\ 10^{-6}$
9	2,818	0,84	3,653	$20\ 10^{6}$	$0,9\ 10^{-6}$
10	3,162	0,84	4,099	$20\ 10^{6}$	$2,6\ 10^{-8}$

Results of calculations are presented on figure 8.1 (curve «Code (31,26)»).
In all energy calculations of systems with coding use as a rule value of the relation of energy of the signal spent for transmitting of one information binary symbol (bit) to power spectral density of noise $h_b^2 = \frac{E_b}{N_0}$ which is considered as *a uniform criterion* of power expenses for an information transfer through the channel with coding and without it.

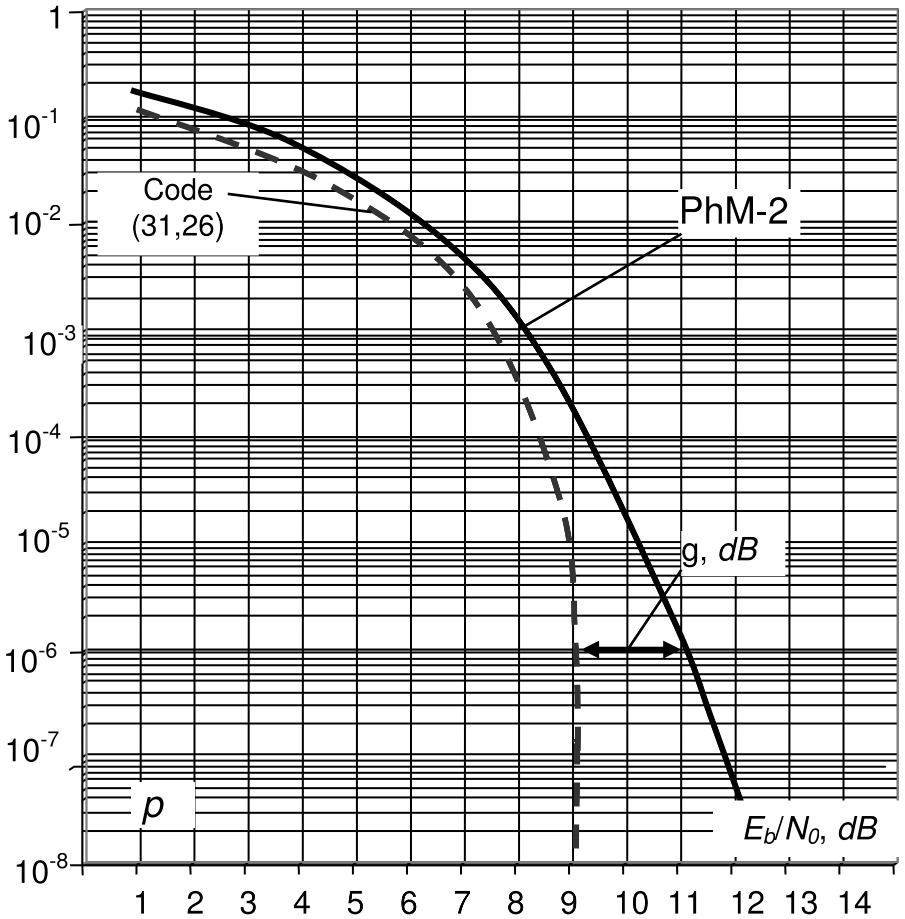


Figure 8.1– Decoding noise immunity of the cyclic code

Size change h_b^2schows *efficiency of application of a error-correcting code*. The effect of errors decrease on a decoder exit can be used on a miscellaneous.
Error-correcting coding provides reduction of the error probability in the received messages. It is well visible from comparison of curves $p=f\,(E_b/N_0)$ on figure. 8.1 for cases of an information transfer by method uncoded PhM-2 and with using of a cyclic code (31,26). It is visible, that by using of an error-corecting code it is possible to admit certain decrease in a channel signal/noise ratio and to receive, accordingly, a *energy gain g* (dB). The value of gain can be defined at various levels of bit error probability *p* on demodulator and decoder exits. Told it is illustrated by the curves of a noise immunity presented on figure 8.1.

> The *energy coding gain* from error-control coding *g* is equal *to a difference* of values E_b/N_0 necessary for maintenance of bit error probability in transmited data by both absence and using encoding.

In particular, for the data resulted on figure 8.1 value gain is g=2,0 dB ($p=10^{-6}$). The gain values *g* is widely used for a choice of codes by designing of telecommunication systems. Values g received at use of cyclic codes in channels with PhM-2 are resulted in table. A.2 of Attachment A.1.

Example 8.1. Optimisation of a cyclic code parametres.

Let's consider the optimisation procedure of the cyclic codes parametres used in the binary symmetric channel with signals PhM-2 for the purpose of the maximum energy coding gain from error-control coding provided by factor of expanding of a signal spectrum band will not exceed K_F=2 (double expansion of a signal frequency band in the channel). Preliminary, under table A.2 Attachment A.1 we will make selection of cyclic codes which can meet requirements on factor of expansion of a band (K_F <2, R_{code}> 0.5). Results of such selection are shown in table 8.3. In table columns values of a code rate are specified. In cells in the lines the gain values (in dB) for various lengths of the code word *n* are presented. Under table A.1 of Attachment A.1 we select the cyclic codes with block length *n*=255 with the rate which is close to optimum rate R_{code}=0,8.

Table 8.3– Parametres of a cyclic codes meeting requirements on a code rate

Word length *n*	Code rate R_{code}			
	0,5	0,6	0,7	0,8
63	2,7	2,8	2,7	2,1
127	3,4	3,5	3,3	2,8
255	3,9	4,0	3,8	3,3

It is visible, that the greatest value of gain g =4,0dB is reached at using enough the long cyclic codes with length of the word n=255. In table 8.4 parametres of the optimum cyclic code are shown.

Table 8.4 – Characteristics of an optimum cyclic code

n	k	$Q_{corr.}$	Code rate R_{code}	Gain g,dB
255	207	6	0,811	4,0

The selected code (255,207) provides a power gain 4,0 dB at rate R_{code}=0,811. Factor band expansion K_F=1,23 not exceeding preset value $K_{F\ (max)}$=2.

Questions

8.1.What parametres of block error-correcting codes define the error probability by decoding in the binary symmetric channel?

8.2. How the energy coding gain is defined?

8.3. What is the reasons of signal frequency band expansion with coding ?

Tasks

8.1. By a technique stated in the Example 8.1 define parametres and generator polynomial of the cyclic code providing the minimum expansion of a signal frequency band by energy coding gain g> 3,0 dB.

8.2*. By data from table A.2 Attachment A.2 construct dependence family of a energy coding gain g from code rate for various lengths of the code word for the cyclic code. Draw conclusions on influence of length of the block on the gain value.

8.3*. By data table A.2 Attachment A.2 construct dependences of a necessary code rate from a demanded energy coding gain g for various lengths of the code word for the cyclic code.. Draw conclusions on influence of a word length on the exchange parities between gain and factor of signal band expansion.

Lecture 9

Structure and characteristics of the convolutional codes

Plan

9.1. Description methods of the convolutional codes [1, Section10.12]
[2, Section 3.4]
9.2. Key parametres and classification of the convolutional codes[2, Section 3.4]

9.1. Description methods of the convolutional codes

Convolutional codes(CC) form a *subclass of continuous codes*. The name *«convolutional code»* occurs that the result of coding on a encoder exit is formed as *convolution* of coded information sequence with pulse responce of the encoder. Encoder of CC contains one or several registers from delay elements and the converter of information sequences into code sequences. Coding process is made *continuously.* The scheme of the simple encoder is shown on figure 9.1.

Information binary symbols ***a*** arrive on an input *of* the register with *K* delay elements *D*. On exits of adders on the Module-2 code symbols ***b*** $^{(1)}$ and ***b*** $^{(2)}$ are formed. Inputs of adders are connected to certain inputs of encoder register elements . The switch *K* on a encoder exit establishes the send sequence of a code symbols to the channel. During one input information symbol it is formed two output code

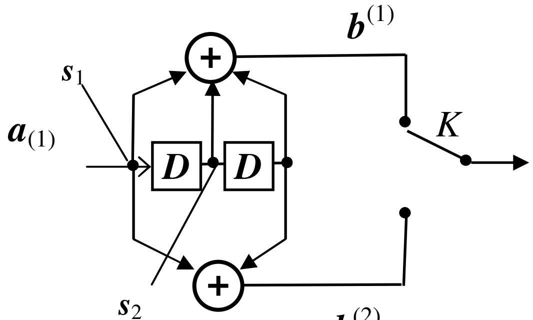
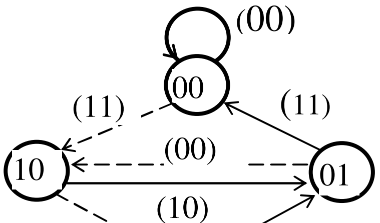


Figure.9.1 – Encoder of CC

Code rate is $R_{kode}=k/n$, where *k*– number of the information symbols simultaneously arriving on inputs of the encoder, and *n*– number of code symbols corresponding to them on encoder exits. Code rate in this example is equal $R_{kode}=1/2$. Coding with other speeds is possible. The convolutiuonal encoder as the *finite state machine* with final number of states can be described by the *state diagramme.* It is considered to be the state as the symbol set on the inputs of register delay elements. For example, symbols (s_1, s_2) designate *encoder state* on figure 9.1. The state diagramme represents the *directed graph* who describes all possible transitions of the encoder from one state into another and also contains encoder output symbols of the which accompany these transitions.

The example of the encoder state diagramme is shown on figure 9.2. It contains four possible encoder states (S_1S_2) =00, 10, 11 and 01 and possible transitions.



Symbols about arrows designate symbols on a encoder output ($b^{(1)}b^{(2)}$), corresponding to the given transition. Continuous lines note the transitions made at receipt on an encoder input of the information symbol *0* and dotted – by the receipt of a symbol 1. Originally the encoder is in a state 00, and receipt on its input of an information symbol a=0 translates it also in a state 00. Thus on an encoder output there will be symbols ($b^{(1)} b^{(2)}$) =00. On the diagramme this transition is designated by a loop "00" leaving a state 00 and again coming back in this state. Further, at symbol receipt a=1 the encoder passes in a state 10 thus on an output there will be symbols $b^{(1)} b^{(2)}$=11. This transition is designated by a dashed line from a state 00 into a state 10. Further, receipt on an input of the coder of information symbols 0 or 1 is possible. Thus the coder passes into a state 01 or 11, and symbols on an output will be 10 or 01, accordingly. Process of a forming of the diagramme comes to an end, when all possible transitions from each state in all the others will be seen. The *trellis diagramme* (*trellis)* is development of the state diagramme in a time. On a trellis the states are shown by knots.The states are connecting by lines. After each transition from one state into another there is a displacement on one step to the right. The example of the trellis diagramme is shown on figure 9.3. The trellis diagramme gives evident representation of all *allowed ways* which are analogues of the allowed code words of a block codes. On them the encoder can move ahead by encoding.

To each information sequence on a encoder input there corresponds a *unique way through a lattice.*

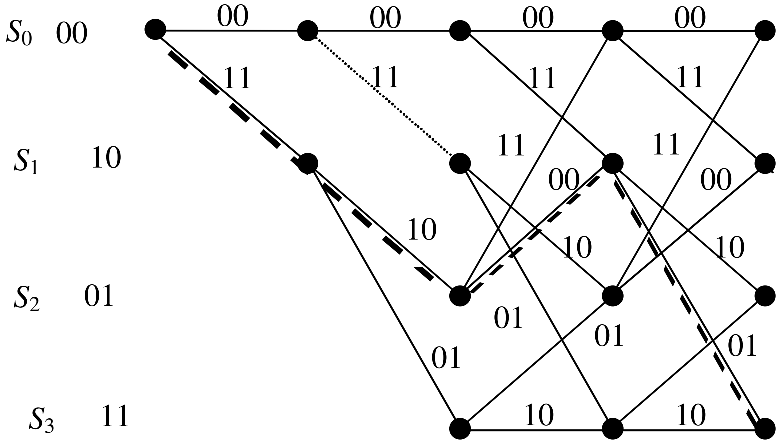


Figure 9.3 – Trellis diagramme

In particular by a dotted line the way on a trellis …11100001… is shown corresponding to an input information sequence …1011.. . For the description of encoder work the sequence of input and output symbols it is convenient to representing with use of the delay operator D in the form of infinite series:

$$\boldsymbol{a}_{(i)}(D)=a_{(i)0}D^0+a_{(i)1}D^1+a_{(i)2}D^2+...,$$

$$\boldsymbol{b}^{(j)}(D)=b_0^{(j)}D^0+b_1^{(j)}D^1+b_2^{(j)}D^2+... .$$

Here indexes in brackets designate:

i– number of an encoder input, $1\le i\le k$;

j– number of an coder output, $1\le j\le n$.

Indexes without brackets (0, 1, 2...) designate the discrete time moments.

For exposition of convolution coding use concept *of a generator polynomial.*

The convolution code will be completely set, if the encoder scheme is known:

– An amount of inputs of the encoder k;

– An amount of outputs of the encoder n;

–Length of each of registers K_i,

– Connections of summators with register cells are specified.

For codes with rate $R=1/n$ the connection of j-th summator ($1\le j\le n$) with cells of the shift register is described by the representation *of a generator polynomial:*

$$\boldsymbol{g}^{(j)}(D)=g_0^{(j)}+g_1^{(j)}(D)+g_2^{(j)}(D^2)+...+g_v^{(j)}(D^v). \qquad (9.1)$$

Here $g_k^{(j)}=1$ if connection *of* j-th summator with k-th register cell exists, and $g_k^{(j)}=0$ if such connection is absent.

Coding process can be presented as multiplication of a generator polynomial $\boldsymbol{g}_{(i)}^{(j)}(D)$, on an input information sequence $\boldsymbol{a}_{(i)}(D)$:

$$\boldsymbol{b}^{(j)}(D)=\boldsymbol{a}_{(i)}(\boldsymbol{D})\boldsymbol{g}_{(i)}^{(j)}(D),\ 1\le i\le \kappa;\ 1\le j\le n. \qquad (9.2)$$

For example, the encoder on fig. 9.1 is characterised by generator polynomials $\boldsymbol{g}^{(1)}(D)=1+D+D^2$ and $g^{(2)}(D)=1+D^2$ or, noting sequence of a factors g_k in the form of *binary combinations*, we receive $\boldsymbol{g}^{(1)}=(111)$ and $\boldsymbol{g}^{(2)}=(101)$. For long codes often use the *octal form.*. In this case generator polynomials will be presented so: $\boldsymbol{g}^{(1)}=(7)$ and $\boldsymbol{g}^{(2)}=(5)$, or $\boldsymbol{G}=(\boldsymbol{g}^{(1)},\boldsymbol{g}^{(2)})=(7,5)$.

Coding process can be described also with using *of* generator matrixes (accordingly, parity check matrixes). It is possible to familiarise with this material under the manual [2, Section 3.4, p. 114] more in detail.

9.2. Key parametres and classification of the convolutional codes

Code rate is defined as

$$R_{\text{code}}=\frac{\boldsymbol{k}}{\boldsymbol{n}}, \qquad (9.3)$$

where *k* – an amount of the information symbols simultaneously arriving on *k* encoder inputs , *n* – an amount of code symbols corresponding to them on *n* encoder outputs.

Use some parametres for definition of *memory length* by coding. The *length of the encoder register* (LCR) *K* is equal to an amount of delay elements containing in the encoder scheme. LCR often apply to memory definition by coding with rate $R_{code}=\frac{1}{\boldsymbol{n}}$, when the encoder contains one register. The encoder represented on figure 9.1 has LCR *K*=3. If the encoder contains some inputs (*k*> 1) so lengths of the registers connected to each input, can be various. In this case define a *code constrained length.*

The *code constrained length* on each input is defined by the *higher degree* of corresponding generating polynomials

$$V_i = max\ [deg\ \boldsymbol{g}_{(i)}^{(j)}\,(\mathrm{D})].$$

The resultant code constrained length is defined by the sum:

$$\nu = \sum_{i=1}^{k} \nu_i \qquad . (9.4)$$

For codes with one register(*k*=1) the values ν and *K* are connected by a simple relation

$$\nu = K. \qquad (9.5)$$

For comparison of a *decoding algorithm complexity* use complexity performance. As it was marked earlier, development of the trellis diagramme consists in a repetion of the same step (see figure 9.3). The *diagramme complexity* is accepted to define an amount of branches on a step of the trellis diagramme. The number of states of a lattice is defined by number of variables *K=v* on inputs of the. register elements. As a result *complexity of one lattice step* can be defined an amount of branches on this step:

$$W=m^{(v+k)} \qquad (9.6)$$

The decoding noise immunity depends on *distance properties* of code sequences on a encoder input . Thus for binary codes more often use distance between sequences estimate in *Hamming metric.*

Free distance of a convolution code d_f – is a minimum distance between two arbitrary semi-infinite sequences on the encoder output which differing from the first branch.

For short codes the free distance can be defined under the state diagramme. If the binary code diagramme is set *free code distance is equally to minimum Hamming weight of a way under the diagramme from a state 00 in the same state* (excepting a loop at this state). On the diagramme figure 9.2 it is visible, that free distance $d_f = 5$.
On the value of free distance judge about *correcting properties* of convolution codes. In particular, if two ways on encoder output, going out from one state on the trellis diagramme, differ in Hamming metric on the value d_f, that by decoding on a minimum distance (with analogy to a case of block codes decoding (see Section 3.1)) *the breavity of corrected errors* is defined by expression

$$. \qquad q_{\text{corr.}} \le \frac{d_f - 1}{2}, \ (d_f\text{– odd}) \qquad (9.7)$$

The free distance is used for an estimation of a noise immunity of convolution codes decoding with decoding algorithms by a maximum aposteriory probability or close to them (*Viterby algorithm* etc.).
In a *systematic code* on k (from n possible) encoder outputs there are information sequences of transmitted symbols, and on remaining *(n–k)* exits – the sequences of the additional symbols formed as *linear combination of information symbols*. By rate R=1/2 generator polynomials of a *systematic code* look like

$$\boldsymbol{g}^{(1)}(D)=1 \text{ и } \boldsymbol{g}^{(2)}(D)=g_0{}^{(2)}+g_{(1)}{}^{(2)}D+g_{(2)}{}^{(2)}D^2+...+g_\nu{}^{(2)}D^\nu.$$

Systematic codes allow to receive on a receiving site an *estimation of information symbols,* without decoding or any other processing of received symbols. *Nonsystematic codes* do not possess such property. As well as in case of a block codes the using of convolution coding with rate $R_{code}=\frac{\boldsymbol{k}}{\boldsymbol{n}}$ leads to e*xpansion of a signal frequency band* in the channel. Thus the of *band expansion factor* is defined by expression:
.

$$K_F=\frac{\boldsymbol{n}}{\boldsymbol{k}}. \qquad (9.8)$$

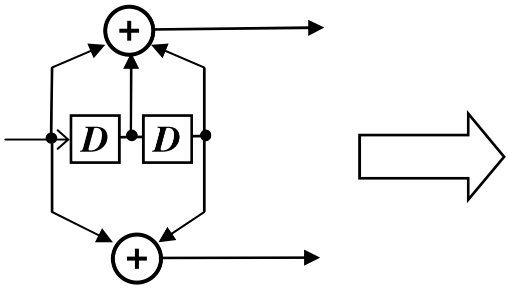
By small code rates the considerable band expansion becomes unacceptable, therefore try to apply encoding with a high code rate. Practically, a choice of code parametres make on the basis of the *compromise*, proceeding from demanded level energy coding gain and admissible value of frequency band expansion factor.

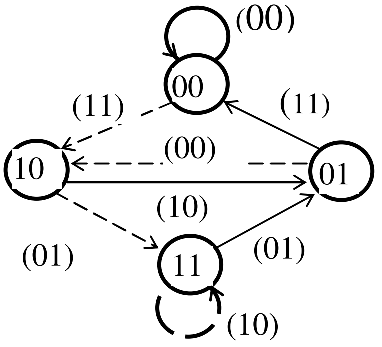
Exercise 9.1. The analysis of code parametres correlations.

Using consecutive modification of the structure of initial encoder (7,5) and corresponding to it state diagramme and a lattice (figures 9.1, 9.2 and 9.3) we will establish correlations of the encoder parametres *k, n, R*$_{\text{code}}$, S, and generator polynomial with code free distance d_f. We will consider some variants of the codes:

1 .Initial code (7,5)

State diagramme





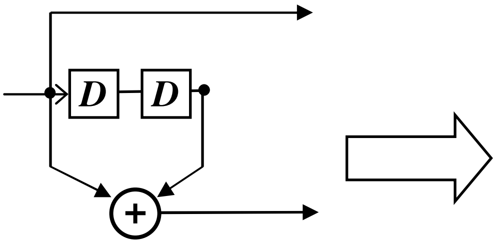
Code parametres: $k=1,$
$n=2,$
$K=2,$
R_{code}=1/2,
*(*for bynary code, m=2) → S=2^K=4,
Free distance d_f=5. Code is nonsystematic.

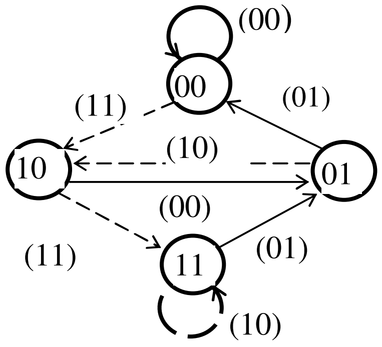
2. Forming of a systematic code (1,5)

Let modify the first polynomial of an initial code, having left one connection, as shown more low in figure. The state diagramme will partially vary. The number of a states remains former as the structure of the encoder register has not varied. Nonzero branches vary: according to a modification of the first generator polynomial on a place of the first numeral of a branch it is necessary to write down the first numeral of a state to which this branch is directed. The code rate also has not varied.

Encoder

State diagramme





Code parametres: $k=1,$
$n=2,$
$K=2,$

R_{code}=1/2,

(for bynary code, m=2) → $S=2^K=4$, The free distance has decreased: d_f=3.

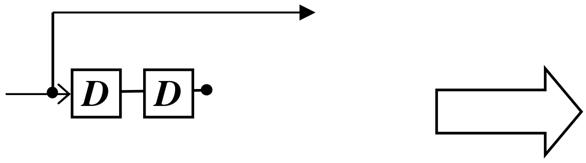
Code is systematic.This example illustrates the general conclusion of a coding theory:

On the free distance the systematic code appear worse of a nonsystematic codes from which they are organised. Therefore *in practice prefer to use the nonsystematic codes.*

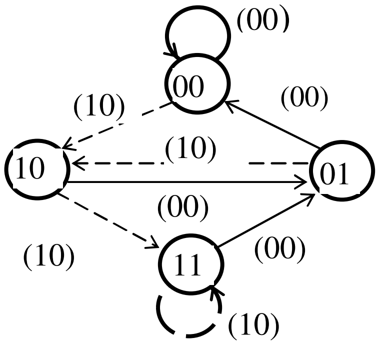
3. Forming of a *nonredundancy code* (1,0).

This, apparently, the "exotic" example allows to reveal a role of the nonzero generator polynomials forming additional symbols.

Encoder



State diagramme



Code parametres: *k=1,*
n=1,
K=2,
R_{code}=1,

(for bynary code, m=2) → $S=2^K=4$,

The free distance has considerably decreased → d_f=1.

Encoder is *systematic* without a additional symbols.

Actually, *nonredundancy coding* is present (*memory of the encoder is not used*). Therefore the code free distance is equally d_f=1, also corresponds to a rate of the *nonredundancy code* R_{code}=1. All increment of free distance in the a code considered in variant 2 spoke presence of nonzero additional symbols.

In Attachment A.3 performances of binary convolution codes with maximum free Hamming distance for various code rates are given.

Questions

9.1. Name key parametres of convolution codes.

9.2. What are construction rules of the state diagramme?

9.3. What connection between the state diagramme and the trellis diagramme?

9.4. How to define a free distance under the state diagramme?

Tasks

9.1. Generator polynomials ($g^{(1)}$,$g^{(2)}$) = (1101,1111) are set. Define parametres such code. What octal and polynomial representations ($g^{(1)}$ (D), $g^{(2)}$ (D)) this code?

9.2**.** Form a functional scheme of code with such set of the generator polynomials.

9.3*.Construct the state diagramme and the trellis diagramme of such code. Show, how on them to define the free distance of a code. Discover a line corresponding to this code in tables СК from Attachment A.3. By analogy with Exercise 9.1 analyse correlation of this code parametres with value of free distance. Make generalising conclusions.

9.4*.Prepare the trellis diagramme of a code (1,5) from the Exercise 9.1, necessary for a analyse of the Viterby algorithm.

Lecture 10

Decoding algorithms of the convolution codes

Plan

10.1. Classification of the decoding algorithms [1, Section 10.12; 2, Section 3.6]

10.2. Viterbi algorithm for decoding of convolution codes [2, Section 3.6]

10.1. Classification of the decoding algorithms

By the receiving for the purpose of optimum solution the received sequence of symbols accepted from the channel is necessary to compare with all possible transmitted sequences. As the number of possible sequences of length N by binary code is equally to 2^N by the big sequence lengths the decoder it becomes inadmissible complexity*(exponential decoding complexity,* see Section 6.3*),* and optimum decoding– practically difficultly realising. However by a big N substantial increasing of transmission fidelity as the noise averages on a long sequence is possible. Therefore the problem of a complexity decreasing of a decoding algorithms is important**.** Two groups of decoding methods for convolution codes are known:

1. *Algebraic decoding methods* are based on the use of algebraic properties of code sequences. In some cases these methods lead to a simple realisations of the codec. Such algorithms are *not optimal,* as used algebraic decoding procedures are intended for correction concrete (and not all) configurations of a channel errors. Algebraic methods identify with *«element-by-element reception»* of sequences which for codes with redundancy, as is known, yields the worst rezults, than *«reception in a whole».* Most simple of algebraic algorithms is *the threshold decoding algorithm.* This algorithm *is so far from optimum* and consequently is seldom used, first of all, in systems with a high information rate. More detailed description of threshold algorithm and its modification can be discovered in the manual [2, Section 3.6.3].

2. *Probability decoding methods* considerably is more near optimal to « *reception in a whole*» as in this case the decoder operates with the values wich proportional to *aposteriory probabilities*, estimates and compares probabilities of various hypotheses and on this basis carries out decision about transmitted symbols.

Algebraic algorithms operate with the limited alphabet of input data for which deriving on an exit of the continuous channel it is necessary to fulfil *quantization* of an received signal with noise. Processes of elaborating of the signals in an exit of the demodulator for antipodal signals are shown on figure10.1 where are presented:

a,c)– forms of antipodal signals in the sampling time on an input of decision device of the demodulator;

b)– binary quantization and transition graph by a *hard decision;*

d)– octal quantization and transition graph by a *soft decision.*

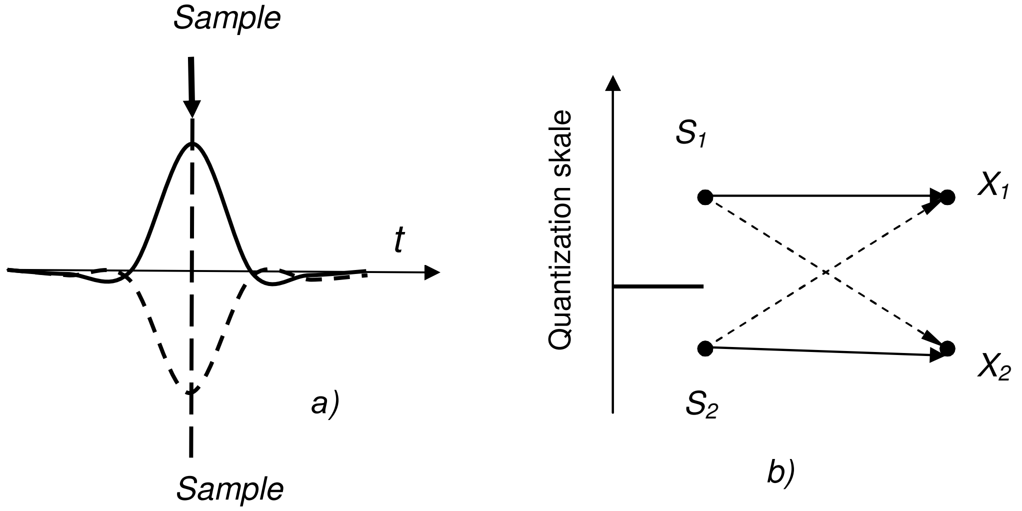
In the simple case make *quantization* of each channel symbol in a *sample time* on two levels (named in the literature as «*a hard decision*»). Thus the hard decision is presented by one binary symbol. It is shown on figure 10.1a,b. By a hard decision number of quantization levels is *L=2*).

By a *soft decision* number of quantization levels is *L*>2(figure10.1d). By a soft decisiont the quantized output describes magnitude of decoded signal plus noise *more precisely* that raises a noise immunity.

Two basic probability algorithms for decoding of convolution codes, and also their various modifications are known.

Sequential decoding algorithm ensures arbitrarily small error probability by a nonzero transmission rate of messages through the channel. By sequential decoding the *search of the way through the code lattice*, corresponding to the transmitted informational sequence is made. Sequential decoding is used for decoding of long convolution codes. The detailed deskription of sequential coding algorithm has presented in the book [4, Section 13.18].-Other variety of probability algorithms is the algorithm based on a principle of a *dynamic programming*, and known as *Viterby algorithm.*

Dynamic programming principle has been formulated in 1940 by R Bellman.
It has wide application in the control theory. In 1970 the dynamic programming in the form of decoding algorithm for convolutional codes has been applied by A.Viterbi to solving of the telecommunication problems (*Viterbi algorithm*).



Viterbi algotithm finds wide application and realises *search of maximum probable way through code lattice* with rejection of a part of the least probable variants of *decoded paths. Viterbi algorithm is characterised by a constant of computing work,* however complexity of decoder Viterbi grows, as by all full search algorithms under the exponential law from code length. Therefore *Viterby algorithm is used for decoding of short convolution codes.*

10.2. Viterbi algorithm for decoding of convolution codes

Let's considerViterbi algorithm on an example of a code with rate $R_{code}=1/n$.
Let, since a moment t=0, on a encoder input is the information sequence of length in L synbols $\boldsymbol{a}_{\boldsymbol{L}} = (a_0a_1\ a_{L\text{-}1})$ moves. On a encoder output there will be a sequence of symbols $\boldsymbol{b}_L = (b_0b_1...\ b_{L\text{-}1})$. An encoder states at the moment t define as a set of ν information symbols $\boldsymbol{w}_{\boldsymbol{t}}=(a_t\ a_{t\text{-}1}\ ...a_{L\text{-}1})$.The trellis diagramme of a code univalently connects the information sequence $\boldsymbol{a}_{\boldsymbol{L}}$, sequence of the encoder states $\boldsymbol{w}_{\boldsymbol{L}}$ and the sequence of the output symbols $\boldsymbol{b}_{\boldsymbol{L}}$. To each branch $\boldsymbol{b}_t$ in the channel there corresponds a signal, which can be presented a set of coordinates $\boldsymbol{S}_{\boldsymbol{t}} = (S_t^{(0)}\ S_t^{(1)}...\ S_t^{(N)})$, where

N– the dimension of a signal space. In the channel the additive noise operates. Then arriving on an decoder input the receiving signal sequence will be equal to
$\boldsymbol{X}_{\boldsymbol{L}}=\boldsymbol{S}_{\boldsymbol{L}}+\boldsymbol{n}_{\boldsymbol{L}}$ where $\boldsymbol{S}_L = (S_0\ S_1...\ S_{L\text{-}1})$ and $\boldsymbol{n}_{\boldsymbol{L}} = (n_0n_1.\ n_{L\text{-}1})$, $\boldsymbol{n}_{\boldsymbol{t}} = (n_t^{(0)}\ n_t^{(1)}...\ n_t^{(N)})$
is N-dimensional vector of a noise.

Decoding consists in *tracing through a code lattice* of a way with a maximum aposteriory probability. It is possible to specify the *decoded way* to one of kinds:
– by the set of estimations of code branches $\mathbf{S}_{\mathrm{L}} = (\mathrm{S}_0\mathrm{S}_1 \ldots \mathrm{S}_{\mathrm{L-1}})$ which making a way;
–by the sequence of estimations of the encoder states $\boldsymbol{W}_L = (w_0, w_1 \ldots . w_{\mathrm{L-1}})$;
– by the sequence of estimations of information symbols on the encoder input
– $\boldsymbol{A}_{\mathrm{L}} = (a_1 \ldots a_{L\text{-}1})$ which coincide with the first symbols of state estimations
– $\boldsymbol{S} = (s_1, \ldots\ s_{\mathrm{t\ -\ v\ +\ 1}})$. The sequence $\boldsymbol{X_L}$ will be decoded with the minimum error probability if from all possible ways to choose estimation $\boldsymbol{S}_{\mathrm{L}}$ for which *aposteriory probability P* ($\boldsymbol{S}_{\mathrm{L}}/\boldsymbol{X}_L$) *is maximum.* Transmission of all variants of sequences $\boldsymbol{a_L}$ considers equiprobable. In this case decoding by criterion of a maximum aposteriory probability is equivalent to decoding by criterion of a maximum of a probability when estimation $\boldsymbol{S}_{\mathrm{L}}$ ensuring performance of condition $P(S_{\mathrm{L}}/X_L)$=max gets out. In the channel without memory conditional probability $P(S_{\mathrm{L}}/X_L)$ is proportional to product of conditional densities of the sum of a signal and a noise:

$$\boldsymbol{P(X_L/S_L)}=\prod_{t=0}^{L-1}\boldsymbol{P}(\boldsymbol{X}_t/\boldsymbol{S}_t)=\prod_{t=0}^{L-1}\boldsymbol{P}(\boldsymbol{X}_t^{(0)}\boldsymbol{X}_t^{(1)}\ldots\boldsymbol{X}_t^{(N)}/\boldsymbol{S}_t^{(0)}\boldsymbol{S}_t^{(1)}\ldots\boldsymbol{S}_t^{(N)})\,.$$

In Gaussian channel by the white noise with an one-sided power spectral density N_0 each factor of this product looks like:

$$\boldsymbol{p}(\boldsymbol{X}_L/\boldsymbol{S}_L)=(1/\sqrt{\pi\boldsymbol{N}_0})^N\exp\{-[\sum_{i=1}^{N}(\boldsymbol{X}_t^{(i)}-\boldsymbol{S}_t^{(i)})^2]/2\boldsymbol{N}_0\}$$

For maxima search we will take the logarithm:

$$\ln\boldsymbol{P}(\boldsymbol{X}_L/\boldsymbol{S}_L)=\ln\prod_{t=0}^{L-1}(1/\sqrt{\pi\boldsymbol{N}_0})^N\times$$

$$\times\exp\{-[\sum_{t=1}^{N}(\boldsymbol{X}_t^{(l)}-\boldsymbol{S}_t^{(i)})^2]/2N_0\}=\boldsymbol{NL}\ln(1/\sqrt{\pi\boldsymbol{N}_0})-\sum_{t=0}^{L-1}\sum_{i=1}^{N}(X_t^{(i)}-S_t^{(i)})^2/2N_0.$$

By the decoding choose sequence of signals $\boldsymbol{S}_{\mathrm{L}} = (S_1, \ldots . S_{L\text{-}1})$ and sequence of branches univalently connected with it $\boldsymbol{S}_{\mathrm{L}} = (S_0\ S_1\ \ldots\ S_{L\text{-}1})$ which *ensures a sum minimum:*

$$\mathrm{MP}=\sum_{\mathrm{t=0}}^{\mathrm{L-1}}\sum_{\mathrm{i=0}}^{\mathrm{N}}(\mathrm{X}_{\mathrm{t}}^{(\mathrm{i})}-\mathrm{S}_{\mathrm{t}})^2=\min,$$

which is called as the *metric of the decoded path*(MP). The path metric contains the *metric of branches*(MB):

$$\mathrm{MB}=\sum_{\mathrm{i\text{-}1}}^{\mathrm{N}}(\mathrm{X}_{\mathrm{t}}^{(\mathrm{i})}-\mathrm{S}_{\mathrm{t}}^{(\mathrm{i})})^2$$

In Gauss channel the *branch metric* is proportional to quadrate Euclidean distances between a vector of the received sum of a signal plus noise $\boldsymbol{X}_t$ and a vector of signal $\boldsymbol{S}_t$ corresponding to a branch of a code $\boldsymbol{A}_{\mathrm{t}}$. In the discrete channel for an estimation of distances use *Hamming metric.* The periodic structure of the trellis diagramme essentially simplifies *comparison* and a *choice of paths* according to decoding rules. The number of a states on a lattice is limited, and two by random chosen enough long paths have, as a rule, the common state. Segments of the paths entering into such states it is necessary to *compare* and *choose* a path with the *least metric.* Such path is called as *survived.* According to Viterbi algorithm such comparison and rejection of segments of path is made *periodically,* on each step of decoding. The simple considering of an example code (7,5) decoding see in manual [2,Section3.6.2, pp.124-127. fig.2.12). According to Viterbi algorithm on each decoding step in the each of trellis states the same type operations are made:

1) *Addition of metrics* of the previous states with metrics of corresponding branches;
2) *Comparison of metrics* of entering paths;
3) *Choice of paths* with the least metrics which values are used as the metric of the states on the subsequent decoding step. If metrics of compared paths are identical, the choice of one of two path is made in a random way.

Realisation complexity of an Viterbi algorithm can be estimated by an amount of branches of the code lattice treated by the decoder at length of decoding *L,* taking into account complexity of each step of a lattice (see formula (9.6)). Complexity of decoder Viterbi realisation can be estimated under the formula:

$$C=m(^{v+k)}\,L \qquad (10.1)$$

On figrure 10.2 the structure scheme of Viterbi decoder intended for work with the demodulator of signals PhM-4 is shown.

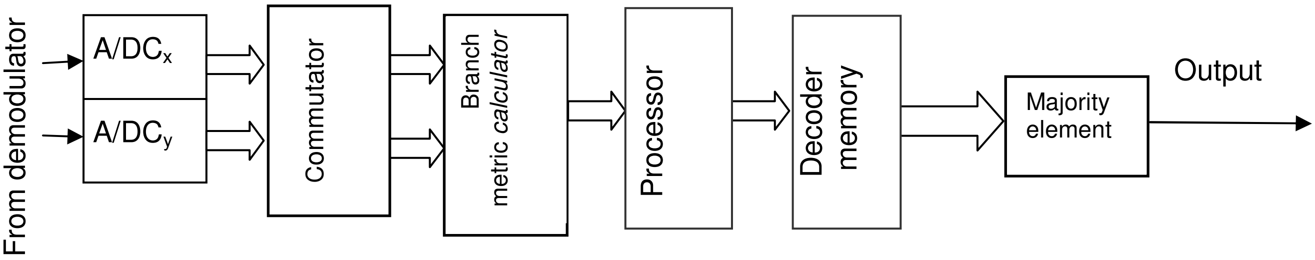


Figure 10.3 – Viterbi decoder structure scheme

The decoder consists from analog/discrete convertors(A/D C) in channels X and *Y*, the calculator of branch metrics , the processor in which operations of addition, comparison and a choice are made, memory device of a survived paths, and majority element in which the path with the greatest metric gets out. The best value of a quantization levels depends on the ratio a signal/noise ont he input A/D C . By eight quantization levels of the losses minimum is ensured at the ratio of a signal magnitude to the quantization step is equal to (4,5... 5,5). More detailed description of assigning and work algorithms of the decoder Viterbi block diagramme elements of are reduced in the manual [2, Section 3.8.2].

Questions

10.1. Whether realisation complexity of the Viterrbi algorithm depends on length of free distance ?

10.2. How will increase complexity of Viterbi decoder by increasing a code costraint length twice?

10.3.* By the expense of what Viterbi decoder complexity raises at using of a soft decision on a demodulator exit?

Tasks

10.1*. Prepare the trellis diagramme of a code (1,5) from Exercise 9.1 which necessary for a illustration of Viterbi algorithm. As free distance of this code d_f=3 (according to formula (9.7) the code corrects single errors) trace decoding process by Viterbi algorithm if in the channel is a single error and establish the fact of its correction by the decoder.

10.2*. Prepare the trellisdiagramme of a nonredundancy code (1,0) from the Exercise 9.1which necessary for a illustration of the Viterbi algorithm. Try to explain by the form the trellis diagramme impossibility of error-correction.

Lecture 11

Noise immunity of convolution codes decoding

Plan

11.1. Decoding error probability of a convolution codes [1, Section 10.12], [2, Section 3.7]

11.2. Energy coding gain [2, Section 3.7]

11.1. Decoding error probability of a convolution codes

The technique of a decoding noise immunity estimation by the convolution codes does not differ from a technique stated in Section 8.1 for a case of block codes. Here too the main role is played code rate R_{code}, code distance properties (in a case of the convolutional codes– the free distance d_f), and decoding algorithm .By using of a decoding algorithm on a probability maximum (Viterbi algorithm) the approximately expression for bit error probability looks like:

$$p \approx \sum_{k=d_f}^{\infty} P_k \ . \qquad (11.1)$$

In this formula P_k – error probability of an way choice on a code lattice o which by transmission of code symbols through a channel with PhM-2 with a white noise with a power spectral density $N_0/2$ is defined so:

$$P_k = 2Q\left(\sqrt{2kR_{code}\frac{E_b}{N_0}}\right), \qquad (11.2)$$

where $Q(z) = \frac{1}{\sqrt{2\pi}}\int_z^{\infty}\exp(-\frac{t^2}{2})dt$ – Gaussian Q-function (a probability integral) which tables of values contain manuals on probability theory and statistical calculations. For practical calculations it is convenient to use enough exact approximation:

$$Q(z) \approx 0.65\, exp[-\, 0{,}44(z+0{,}75)^2]. \qquad (11.3)$$

Evaluations under formulas (11.1) and (11.2) show, that in the sum (11.1) by a big ratio signal/noise the first member (by k=d_f) has the greatest value, and remaining members of the sum with growth k fast decrease. Therefore in practice are limited to use of the simplified formula:

$$p \approx 2Q(\sqrt{2d_f R_{code} \frac{E_b}{N_0}}), \qquad (11.4)$$

As well as by the block coding, comparison of a decoding noise immunity can be made with a noise immunity of coherent receiving of signals with binary phase modulation PhM-2). Thus the calculation formula for bit error probability can be received from expression (11.2) having supposed k=1, R_{code}=1:

$$p_{\text{PhM-2}} = 2Q(\sqrt{2\frac{E_b}{N_0}}), \qquad (11.5)$$

where $h^2 = \frac{E_b}{N_0}$ – the ratio of the signal energy expended on transmission of bit E_b to a power spectral densityof a noise N_0 on an input of the demodulator.

Exercise 11.1. The analysis of a decoding noise immunity

Let's make a calculations of a bit error probability on exits of the demodulator of signals PM-2 and Viterbi decoder included after it, using formulas (11.5) and (11.4) for next codes :

1. Code (5,7), R_{code}=1/2, d_f=5, v=2;
2. Code (133,171), R_{code}=1/2, d_f=10, v=6.

The calculation results are given in table 11.1 and presented on figure 11.1. In the table given values of argument z are specified function Q (z), used in formulas.

11.2.Energy coding gain

As well as by an estimation of a decoding noise immunity of the block codes (see Lecture 8) in a case of convolution codes use concept of an *energy codihg gain.*

The energy coding gain g is equal to a difference between of values E_b/N_0 necessary to get the given error probability p. by the absence and by the coding use.

Values of error probability level at which the gain is defined depends from the requirements to fidelity of the transmitted digital information. For digital telephony systems a demanded level of a bit error probability usually makes p = (10^{-5}... 10^{-6}). In systems of digital TV transmission try to ensure p = (10^{-10}... 10^{-11}).

Table 11.1– Calculation of a decoding noise immunity

E_b/N_0, dB	Bit error probability on an demodulator PhM-2 output		Bit error probability on an decoder output			
			Code (5,7)		Code (133,171)	
	z	$p_{\text{PM-2}}$	z	p_{code}	z	P_{code}
2	1,78	8 10^{-2}	2,815	5 10^{-3}	3,981	7 10^{-5}
3	1,998	5 10^{-2}	3,159	1 10^{-3}	4,467	8 10^{-6}
4	2,241	2,5 10^{-2}	3,544	4 10^{-4}	5,012	5 10^{-7}
5	2,515	1,2 10^{-2}	3,976	8 10^{-6}	6,31	2 10^{-8}
6	2,822	4,7 10^{-3}	4,462	6 10^{-7}	7,079	4 10^{-10}
7	3,552	4 10^{-4}	5,617	2 10^{-8}	7,943	4 10^{-15}
8	3,986	7 10^{-5}	6,302	4 10^{-10}	8,913	1 10^{-18}
11	5,018	8 10^{-6}	7,071	2 10^{-12}	10,0	1 10^{-28}

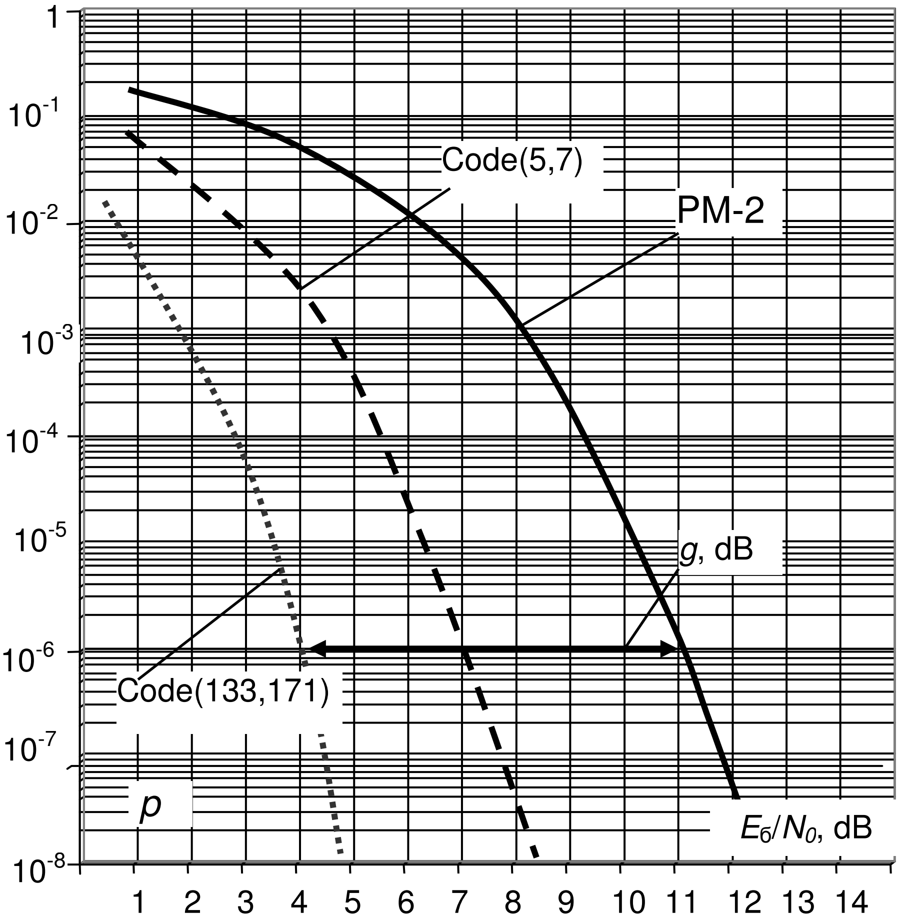


Figure11.1– Decoding noise immunity

The value of coding gain at the given bit error probability p^* can be defined bycomparing the arguments of function Q (z) in a formulas for error probability (11.4) and (1.5) for identical probabilities $p_{code}=p_{\mathrm{PhM\text{-}2}}=p^*$. Calculations show, that gain depends from level of error probability p * on which it is defined. It is well visible on the curves figure 11.1 representing calculation results from Exercise 11.1. Value of a gain with decreasing of a probability p^* aspires to the limit which in the coding theory name as *asymptotic coding gain*:

$$\text{A-gain}=\lim g(p^*\rightarrow 0). \qquad (11.6)$$

Comparing arguments in the expressions (11.5) and (11.4) we come to wide used in energy calculations of telecommunication systems to expression for A-gain in logarithmic units:

$$\text{A-gain}=10\lg(R_{code}d_{\mathrm{f}})(\text{dB}). \qquad (11.7)$$

As A-dain is upper bound of a gain g for fast comparison and a choice of codes use A-gain. Values of this A-gain often include in the code tables (see tables of Attachment A.3). In table 11.1 for an example data about convolution codes with various lengths of a code length v and rate R_{code} are cited. Values of a A-gain are shown. More detailed data are given in tables (A.3 … A.6) from the Attachment A.3.

Table 11.1 – Characterustics of a convolutional codes

Code rate R_{code}	Code constraint length v=4		Code constraint length v=6	
	Code	A-gain, dB	Code	A-gain, dB
1/3	25,33,37	6,02	133,145,175	6,99
1/2	31,33	5,44	133,171	6,99
2/3	31,33,31	5,23	133,171,133	6,02
3/4	25,37,37,37	4,78	135,163,163,163	6,73

Comparison of a gain values ensured by the cyclic coding (see table 8.3 and figure 8.1) with similar parametres for convolutional codes (see table 11.1 and figure 11.1) shows, that convolution codes in a combination to Viterbi decoding algorithm ensure considerably more gain in comparison with block codes. It explains wide using of convolution codes in telecommunication systems for a noise immunity increasing. Typical a using of the code (133,171) ensuring A-gain=6,99 dB by the rate R_{code}=0,5 here is, i.e. at two-multiple expansion of a frequency band of the coded signal. The codecs of such code are developed in the form of the big chips are serially emitted.

Qestions

11.1. How depends the gain from the code constraint length ?
11.2. How depends the gain from the code rate ?

Tasks

11.1. Using tables of a convolutional codes from the Attachment A.3 construct the dependence of a gain from a code rate by the fixed values a constraint code length. Explain tendencies of a gun of these dependences.
11.2. Using tables of the Attachment A.3 choose a codes, ensuring A-gain> 6dB and specify parametres of these codes.

Lecture 12
Increasing of the efficiency of a digital telecommunication systems

Plan

12.1. The theory of efficiency byA. Zjuko. Information, energy and frequency efficiency of a telecommunication systems [1, Section13]
12.2.Limiting efficiency of a telecommunication systems and Shannon bound [1, Section13]
12.3. Perspective ways the further increasing efficiency [1]

12.1. The theory of efficiency byA. Zjuko. Information, energy and frequency effeciency of a telecommunication systems

Generally the result of work of a telecommunication systems is defined by an quantity and quality of the transmitted information. The quantity is estimated by an information transmitting rate through a channel R_{chan} (bit per second), and quality – by the values of an error. According to Shannon theorems, the error with a corresponding choice of a transmission method (i.e. modulation/coding) can be made arbitrarily small (see explicitly the materials of the Module 2). At the same time, the transmission rate cannot be above some informational resource named a channel capacity C_{chan}. . A.Zjuko has suggested to consider as one of indicators of a *system effeciency* the value of mean rate R_{chan} *at* which the given fidelity of an information transfering is ensured. Thus the *information system effeciency* as degree of use of a channel capacity of the channel is defined by relative value $\eta = \frac{R_{chan}}{C_{chan}}$. In real conditions the indicator η *always is less than unit*. The more close η to unit, the more absolutely transmitting information system.

Reaching necessary for a transmission rate and fidelity is accompanied by certain expenditures of other *major resources*: *signal power* P_s and a *channel frequency band* F_{chan}. Such approach has allowed to introduce the indicators: *power (* $\beta = \frac{R_{chan}}{P_s / N_0}$) and *frequency efficiency* ($\gamma = \frac{R_{chan}}{F_{chan}}$), uses of the mentioned resources characterising degree**.** Here P_s/N_0 – the ratio of a signal power to a power spectral density of noise on a receiver input). Thus, *efficiency indicators* by G.Zjuko look like:

– *Information efficiency* of a system which define the degree of a channel capacity using

$$\eta = \frac{R_{chan}}{C_{chan}}; \qquad (12.1)$$

– *Energy efficiency* $$\beta = \frac{R_{chan}}{P_s / N_0}; \qquad (12.2)$$

– *Frequency efficiency* $$\gamma = \frac{R_{chan}}{F_{chan}}\ . \qquad (12.3)$$

12.2. Limiting efficiency of telecommunication systems and Shannon bound

Indicators β and γ make sense a *specific rates*, and inverse values $\beta' = 1 / \beta$ and $\gamma' = 1 / \gamma$ define *specific expenses* of corresponding resources on an information transfering with an unity rate (1 bit per second). For the Gaussian channel with frequrncy band F_{chan}, the ratio of a signal to noise $\rho = P_s/P_n$ and channel capacity $C_{chan} = F_{chan} \log(\rho + 1)$ it is possible to establish, that these efficiency indicators are connected by the relation:

$$\eta = \frac{\gamma}{\log(1 + \frac{\gamma}{\beta})} \text{ и } \gamma = \rho\beta \qquad (12.4)$$

For ideal system (η =1) limiting equation can be defined. According to the

Shannon theorem by the corresponding transmission methods (coding and modulation) and receiving (demodulation and decoding),the value η can be as much as close to unit. Thus the error can be made as much as small. In this case by a condition $\eta=1$ it is received *limiting equation between* β and γ:

$$\beta = \frac{\gamma}{2^{\gamma}-1} \qquad (12.5)$$

This formula defines of energy efficiencyfrom the frequency efficiency for the *ideal system* ensuring *equality of a information rate to a channel capacity.* It is convenient to represent this equation in the form of a curve on a plane $\beta=f\,(\gamma)$ (figure 12.1, a curve « Shannon bound»). This curve is limiting and reflects the *best interchanging between β and γ*in the continuous channel (C-Chan.*)*.

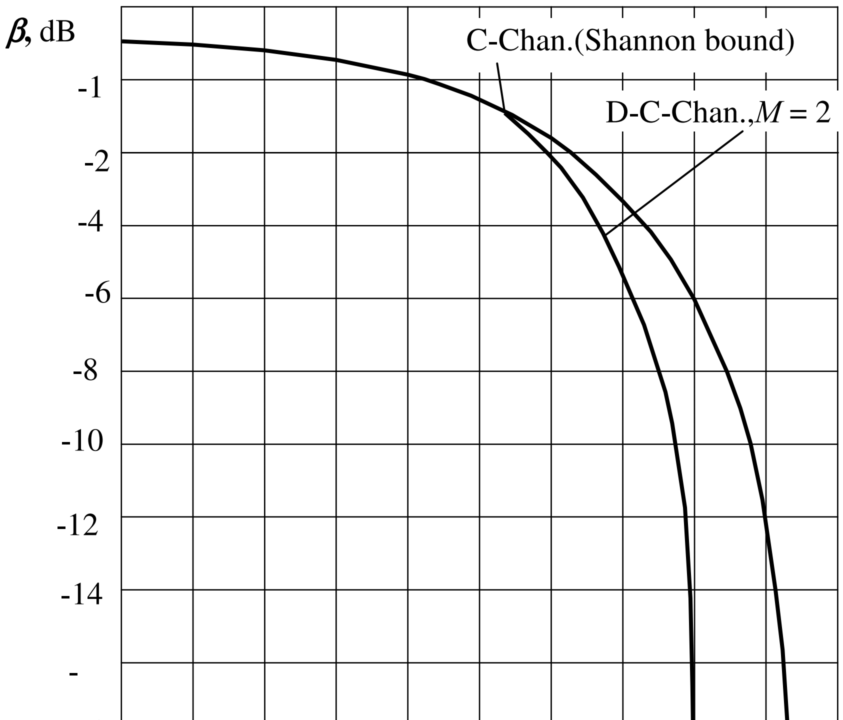
It is necessary to notice, that frequency efficiency γ varies in limits from 0 to ∞ *energy efficiency is bounded above* by magnitude:

$$\beta_{\max} = \lim_{\gamma\to 0}\beta = \lim_{\gamma\to 0}\frac{\gamma}{2^{\gamma}-1} = \frac{1}{\ln 2} \approx 1{,}443 \qquad (12.6)$$

Differently, *energy efficiency* of any information transmitting system ina Gauss channel *can not exceed the magnitude*

$$\beta_{\max} = 1{,}443 \qquad (12.7)$$

Similar limiting curves can be constructed and for any other channels if in formulas (12.2) and (12.3) instead of a rate R_{chan} to substitute expressions for a channel capacity of the corresponding channel. So, in particular, on fig. 12.1 the curve for limiting equation $\beta=f\,(\gamma)$ the is discrete-continuous channel (D/C-Chan.) is shown. It "is enclosed" in a curve of the continuous channel (C-Chan.) that confirms knownresult of an information theory according to which DN channel capacityof D/C-Channel always is less a than channel capacity of the continuous Channel(C-Chan.) which is a basis for construction of corresponding D/C-Channel. In real digital systems error probability p always has a final value and informational efficiency is less then a limiting value $\eta_{\max}$. In these cases for the fixed error probability *p=const* it is possible to define efficiency ratio β to γ and to construct curves $\beta = f\,(\gamma)$.



In coordinates (β, γ) to each variant of a telecommunication system there will corresponds a point on a plane**.** All these points (curves) *should place below a limiting curve* of «Shannon bound». The place of these curves depends on an aspect of signals (modulation), a codes (a coding method) and a method of the elaborating of a signals (demodulation/decoding).

> About perfection of the digital telecommunication methods judge on a degree of placing of real efficiency of to the limiting values.

Concrete data about the efficiency of various modulation/ encoding methods and also their combinations are given in following section.

12.3. Perspective ways of the further increasing efficiency

Using the various methods of the error-correcting coding considered in these Module the designer of telecommunication system owning art of optimisation can flexibly change of the efficiency indicators approaching them to the limiting, potentially possible values which are established in the previous section.

Efficiency of the digital telecommunication systems of transmission can be essentially increased by the application of multi-position signals and error-correcting codes, and also their combinations. The choice of signals and codes in these cases is defining for construction *highly effective codems* (the codecs matching among themselves and modems). Comparison of efficiency of systems with multi-position signals and error-correcting codes is convenient for making with using of

The diagramme $\beta = f\ (\gamma)$, presented on figure 12.1. Thus degree of perfection of a modulation/ coding methods and can be estimated, comparing efficiency with limiting values. Results of the efficiency analysis are presented on figure 12.2. At the same time, comparison various modulation/coding methods is convenient for making comparison taking for "*reference point*" the efficiency of telecommunication system with modulation PhM-4 *(*without error correcting coding*).* From among simple methods it is the most effective and widely used method of modulation/coding with indicators $\gamma = 2$, $\beta = -9{,}6$dB, $\eta \approx 0{,}47$. Conveniently as well that the point representing on figure 12.2 values of efficiency PhM-4 is arranged in a central part of the diagramme**.** If an origin of coordinates to transfer to a point corresponding PhM-

4, in a new frame (*Δβ, Δγ)* on a vertical axis the *power gain* Δβ in comparison with PhM-4, and on a horizontal axis a *gain on a specific rate* **Δ*γ*** will be counted. Let's notice, that all possible modulation/coding methods can be divided into four groups corresponding to four quadrants on the diagramme *β =f (γ)*:

Quadrant III in which are arranged the low efficiency methods having rather PhM-4 *loss on β and γ.*

Quadrant II including methods *with high energy efficiency*, ensuring a gain *on β* in exchange for loss on *γ* (*systems with error-correcting codes*)

Quadrant IV including modulation methods ensuring *a gain on γ in exchange for loss on β* (*systems with multi-position Ph-M and APh-M signals*);

Quadrant I including *perspective modulation/coding methods* ensuring a *simultaneous gain both energy and frequency efficiency.*

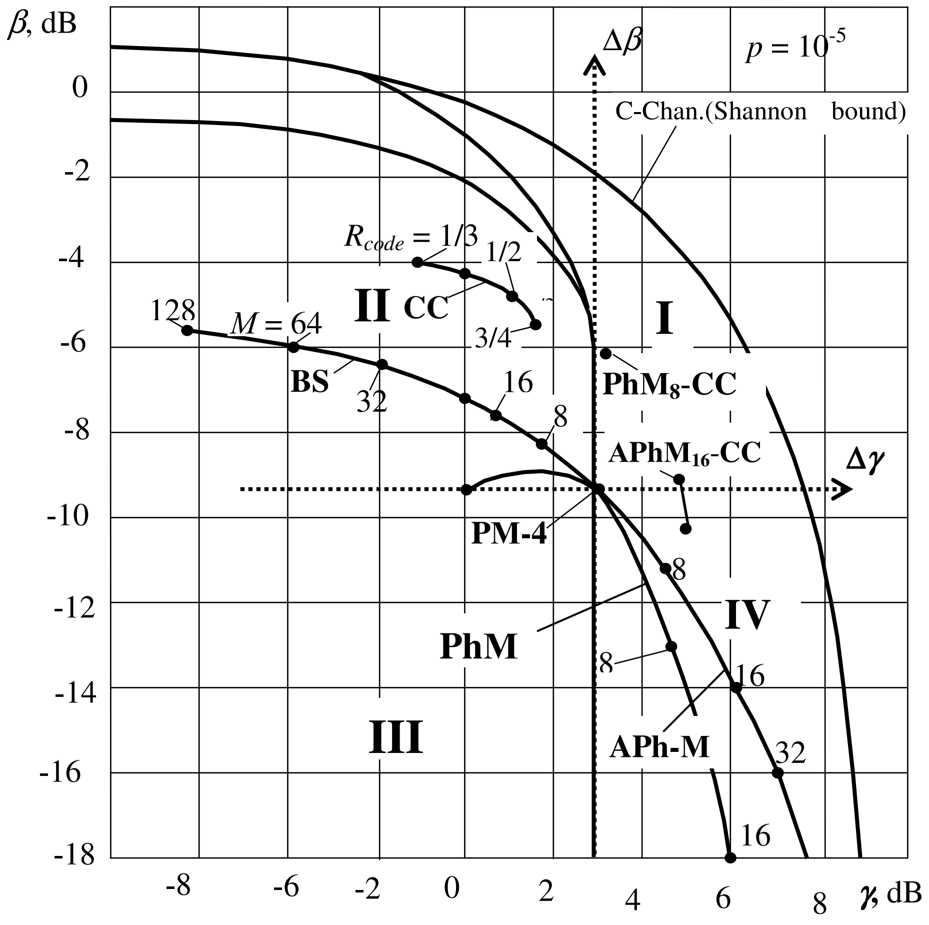


Figure 12.2 – Efficiency of multi-position signals and error-correcting codes

Outcomes of calculations show (figure 12.2, Quadrant I), that application such Signal-to -Code structures allows to receive simultaneously a *gain both in energy, and in frequency efficiency* and, anyway, to get a gain on one indicator, not worsening another. So, system PhM$_8$-CC by the using of a convolution code with R_{code}=2/3 ensures a energy gain $\Delta\beta$=2,8 dB without a decreasing of a specific rate γ, and system APhM$_{16}$-CC by R_{code}=1/2 and ν=3 a gain on a specific rate $\Delta\gamma$=2 dB without a drop of energy efficiency β. Information efficiency of these systems is $\eta\approx$ (0,6 … 0,7). The detailed analysis such *Signal-to -Code structures* is reduced in manual [3 Section 9.2] .

Microelectronic reachings last decade initiated attempts to realise the potentially possible efficiency, despite of growth of decoding complexity. In 1993 *turbo-codes* have been offered. Turbo-codes has been in details in a manua[3, Section 11.1] described. Intensive development of mobile telecommunication systems havs led to the invention of a *time/space coding*, in details described in the manual [3, Section 11.2].

Attachments

A. PERFORMANCES OF ERROR-CORECTING CODES

A.1. Performances and generator polynomials of cyclic codes for channels with opposite signals

In table A.1 the short table of performances and generator polynomials of binary cyclic codes is presented. Generatior polynomials of codes are given in the octal form where:

n–word length;	R$_{code}$–code rate;
k– amount of information symbols in the word;	q_{corr}– breavity of corrected errors.

Example A.1. Octal representation of generatior polynomials.

The code with parametres *n*=7**,** *k*=4, q_{corr}=1 has a polynomial (13) ⇒ (001.011) ⇒ (1011) ⇒ $+x^3+x+1$.

Table A.1 – Performances and generator polynomials of the cyclic codes

n	*k*	q_{corr}	R$_{code}$	Generator polynomials
7	4	1	0,57	13
15	11	1	0,73	23
	7	2	0,47	721
	26	1	0,84	45

	21	2	0,68	3551
	16	3	0,52	107657
	11	5	0,35	5423325
63	57	1	0,9	103
	51	2	0,81	12471
	45	3	0,71	1701317
	39	4	0,62	166623567
	36	5	0,57	1033500423
127	120	1	0,95	211
	113	2	0,89	41567
	106	3	0,84	11554743
	99	4	0,78	624730022327
255	92	5	0,97	435
	239	2	0,94	267543
	231	3	0,91	156720665
	223	4	0,87	75626641375

A.2. Energy coding gain by using of the cyclic codes

In table A.2 the values of energy coding gain *g* (dB) are given for using of cyclic codes in channels with PhM-2.

TableA.2– Energy coding gain g (dB) by using of the cyclic odes

Block length, *n*	Code rate R_{code}					
	0,3	0,4	0,5	0,6	0,7	0,8
31	1,2	1,6	1,9	2,0	1,9	1,6
63	2,0	2,4	2,7	2,8	2,7	2,1
127	2,6	3,1	3,4	3,5	3,3	2,8
255	3,2	3,6	3,9	4,0	3,8	3,3

A.3. Performances of binary convolution codes

In tables (A.3 … A.6) performances of binary convolution codes with maximum free Hamming distance and rates (1/8 … 1/2) are given. Generator polynomials are given in the octal form.

Labels: v– code constraint length; d_f– free Hamming distance; A-gain– asymptotic coding gain (dB) by using code in a channels with Ph-M-2.

Table A.3 – Code rate R_{code}=1/8

Code number	ν	Generator polynomials	$d_{\text{f}m}$	d_f	A-gain, dB
1	4	25,27,33,35, 37,25,33,37	32	32	6,02
2	5	45,55,57,65, 67,73,77,47	36	36	6,53
3	6	115,127,131, 135,157,173, 175,123	40	40	6,99

TableA.4– Code rate R_{code}=1/4

Code number	ν	Generator polynomials	d_{fm}	d_f	A-gain, dB
4	2	5,7,7,7	10	10	3,98
5	3	13,15,15,17	13	13	5,12
6	4	25,27,33,37	16	16	6,02
7	5	51,55,73,77	18	18	6,53
8	5	53,67,71,75	18	18	6,53
9	6	135,135,147,163	20	20	6,99
10	7	235,275,313,357	22	22	7,40
11	8	463,535,733,745	27	27	8,29

TableA.5 – Code rate R_{code}=1/3

Code number	ν	Generator polynomials	d_{fm}	d_f	A-gain, dB
12	2	5,7,7	8	8	4,26
13	2	5,6,7	8	7	3,68
14	3	13,15,17	10	10	5,23
15	3	11,15,17	10	9	4,77
16	3	10,15,17	10	8	4,26
17	4	25,33,37	12	12	6,02
18	5	47,53,75	13	13	6,37
19	5	47,55,75	13	13	6,37
20	5	45,55,75	12	13	6,42
21	6	133,145,175	15	15	6,99
22	6	127,155,165	15	13	6,37
23	7	255,331,367	16	16	7,27
24	8	557,663,711	18	18	7,78

Table A.6 – Code rate R_{code}=1/2

Code number	ν	Generator polynomials	d_{fm}	d_f	A-gain, dB
25	2	5,7	5	5	3,98
26	3	15.17	6	6	4,77
27	3	13,15	6	6	4,77
28	4	23,35	8	7	5,44
29	4	31,33	8	7	5,44
30	4	25,37	8	6	4,77
31	5	53,75	8	8	6,02
32	5	61,73	8	8	6,02
33	5	43,75	8	8	6,02
34	5	45,73	8	8	6,02
35	5	71,73	8	8	6,02
36	6	133,171	10	10	6,99
37	6	135,163	10	10	6,99
38	7	247,371	10	11	6,99

B. Methodical instructions and the input data

for the course work

«OPTIMISATION METHODS OF A NOISE IMMUNITY CODING FOR TELECOMMUNICATION SYSTEM »

Introduction

In Lecture 12 are shown, that noise immunity coding is an effective mean for the optimisation of telecommunication systems. In practice engineer-desiner should solve *optimisation problems* on the basis of numerical calculations and corresponding comparison of a coding methods and a choice of concrete coding methods and corresponding to them the codes. The solution of such problem is necessary in basis of the course work.

Input data are set in the table of variants (table B.2):

1. The digital information is transmitted by a binary code. Kinds of the transmitted digital information are:
   CD– Computer data
   DT– Digital telephony
   DTV – Digital TV

DB – Digital sound broadcasting.

2. The telecommunication channel is the channel with constant parametres and the additive white noise

3.Ratio S/N on a demodulator input $h^2_0=E_b/N_0$

4.Modulation methods: PhM-2, PhM-4

5.Reception – coherent

6. Information source productivity R_{source} (bit per second)

5. Channel frequency band F_{ch} (kHz)

6. Bit error probability less p

7. Admissible complexity of a code lattice – no more W.

It is necessary:

1. To choose and justify a choice of a error-correcting code for projected system, ensuring demanded bit error probability level p under condition of a following ***restrictions:***

1.1.The frequency band of the coded signal F_{code}.should not exceeds of a channel frequency band F_{ch} ($F_{code}< F_{ch}$)

1.2. By using of convolution codes the code lattice complexity should be no more magnitude W

2.To develop and give detailed exposition structural and function schemes of the encoder and the decoder for the chosen code and to justify their parametres

3. 3.To analyse of energy and frequency efficiency of a projected telecommunication system and to compare them to limiting values of efficiency

4. To make a conclusion on the done work.

Content of a project ***explanatory note:***

1. The introduction and input data

2. Exposition of the block diagramme of projected telecommunication system with instructions about places of inclusion of the error-correcting encoder, modulator, demodulator and the decoder with detailed explanations of functions fulfilled by them

3. Classification the error-correcting codes by structure. The comparative analysis of advantages and shortages of noise immunity block and convolution codes. An application substantiation in the project of convolution codes.

4. Classification and the comparative analysis of decoding algorithms for convolution codes. A substantiation of a Viterbi algorithm choice for decoding of a convolutional code

5. Calculation of a spectrum frequency band occupied with a coded digital signal

6. Calculation of values of a spectrum frequency band occupied with the coded digital signal F$_{code}$ depending from a code rate

7. Definition of an admissible code rate R^*_{code} by a condition 1.1 ($F_{code}< F_{ch}$)

8. Definition of the enumeration of codes with the rates which are not exceeding admissible rate R^*_{code} which can be used for a task in view solution

9. Choice codes from this enumeration ensuring given bit error probability level by the Condition1.1 and restriction satisfying to the requirement on decoder complexity (by Condition1.2)
10. Checking calculation of a bit error probability for the decoding of the chosen code
11. Elaborating and exposition of a structural and function schemes of the encoder and the decoder chosen code.
12. An inference with summarising of the performed work
13. The list of the used literature.

Methodical instructions

Calculation of a spectrum frequency band for a signals PhM-2 (PhM-4) should be made under recommendations from the Module 1. Using of an error-correcting codes with code rate R_{rate} leads to increasing of a occupied frequency band of the coded signal in $K_F = 1/R_{\text{code}}$ time (see Lecture 9).

On the other hand, a correcting ability of a code increases with a decreasing of a code rate. Therefore the problem of a code parametres optimisation consists in a choice of a code with a rate at which the frequency band of the coded signal does not exceed the given channel frequency band F_{ch}.. If the demanded channel band for the transmission of a coded PhM-M signal with information rate R_{source} is equal $F_{(\text{Ph-M})}$, and the code rate is chosen equal R_{code} the channel frequency band which is necessary for transmission a coded Ph-M signal will be equal

$$\mathrm{F}_{\mathrm{ch(PhM-code)}} = \frac{\mathrm{F}_{\mathrm{(Ph-M)}}}{\mathrm{R}^*_{\mathrm{code}}}.$$

Then from a unequality ($\mathrm{F}_{\mathrm{ch(PhM-code)}} < \mathrm{F}_{\mathrm{ch}}$) it is received a simple condition for a choice of a code rate:

$$\mathrm{R}_{\mathrm{code}} > \mathrm{R}^*_{\mathrm{code}} \qquad \text{(B.1)}$$

The told is illustrated by figure B.1. The expanding of a frequency band of a coded PhM signal is proportional to factor of a band expansion. By the process of a decreasing of a code rate (increasing of K_F) the frequency band extends and reaches values of a given channel frequency band F_{ch}. On the same figure the line A-gain=f(K_F)) is shown. Intersection of a curve band with boundary given value $\mathrm{F_k}$*defines the admissible factor of a channel frequency band expansion $K_F=1/R_{code}$ and, accordingly, of code rate R^*_{code}.

The first stage of a choice of a error-correcting code is the choice of a class of codes (a class block or convolution codes). Using materials of Lecture s 8 and 11 it is recommended to justify with deep arguments for a choice of a convolution codes for using in the project. Among decoding algorithms on a latitude of practical application the in the lead place occupiesViterbi algorithm. It is recommended to apply Viterbi algorithm in the project. In section of the project with a substantiation of application of this algorithm it is necessary to give information about realisation complexity of an algorithm. Among the codes selected by criterion of a rate according to the formula (B.1) there can be codes with various length of code constraint length (and, accordingly, with various decoder complexity). The noise immunity of decoding is characterised by A-gain. In code tables of a values A-gain are not reduced at certain level of error probability. At the same time, magnitude of A-gain is upper estimation of a real gain. Therefore at a choice of a codes it is recommended to use A-gain which values are available from the Attachment A. Among the selected candidates of codes it is necessary to apply a code ensuring maximum A-gain and meeting maximum requirements on a code rate and minimym complexity of the decoder.

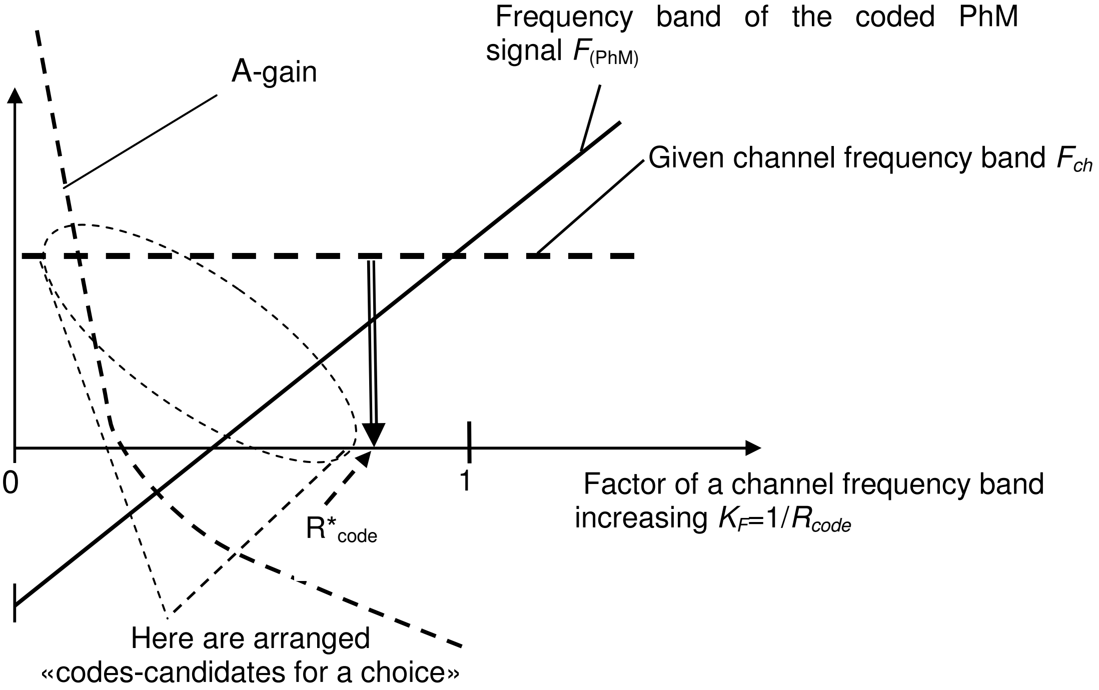

Frequency band of the coded PhM signal $F_{(PhM)}$
A-gain
Given channel frequency band F_{ch}
0
1
Factor of a channel frequency band increasing K_F=1/R_{code}
R*$_{code}$
Here are arranged «codes-candidates for a choice»

Figure B.1– Procedure of a code optimisation

Definitive data about error probability on a decoder exit it is necessary to get by the calculations th [illegible] from the ratio a signal/noise. In case of representation failure to meet requirements it is recommended to apply a code with moore value of A-gain.

Example of calculations and code optimisation procedure

Input datas:

1. Kind of transmitted digital information:
   CD– Computer data
2. Ratio *S/N* h^2_0=4 dB
3. A modulation method PhM-4
4. Reception - coherent
5. Information source productivity R_{source}=64 kbit/s
6. Channel frequency band F_{ch}=100 kHz
7. Admissible bit error probability less p=10^{-5};

8. Admissible code lattice complexity no more *W=150*.

1. Calculation of a necessary channel frequency band for transmission with the method PhM-4 is made under formula $F_{\text{(PhM-4)}} = [R_{\text{source}}(1 + \alpha)]/2$, where α– spectrum roll- off factor. Being set by value α=0,4, we receive

$$F_{\text{(PhM-4)}} = [R_{source}(1 + \alpha)]/2 = [64\ (1+0.4)]/2 = 44{,}8 \text{ kHz}.$$

2. According to the formula (B.1) it is defined limiting value of code rate

$$R^{*}_{\text{(code)}} = \frac{F_{\text{(Ph-M)}}}{F_{\text{ch}}} = \frac{44{,}8}{100} = 0{,}448.$$

3. Under code tables we select the codes, satisfying to the requirement on a rate. Data about these codes are shown in тable B.1.

From the table it is visible, that for the solving given task can be used codes with the rate R_{code}=1/2 which ensure enough big A-gain. In the table data the code with generator polynomials (133,171) which at rate R_{code}=0,5 ensures A-gain =6,99 dB is chosen for the project.

Data of bit error probability calculation is given on figure B 1. It is visible, that the using of a such code ensures such performance:by the ratio signal/noise $h^2{}_0$=4 dB the bit error probability is less than 10^{-5}. Comparison with curves for uncoded PhM- 4 shows that by $p=10^{-5}$ this code ensures A-gain=6 dB.

TableB.1– Performances for a code choice

Code rate R_{code}	Generating polynomials	Code length *v*	Lattice complexity *W*	A=gain dB
1/8	25,27,33,35, 37,25,33,37	4	32	6,02
1/8	115,127,131, 135,157,173, 175,123	6	128	6,99
1/4	25,27,33,37	4	32	6,02
1/4	463,535,733,745	8	512	8,29
1/3	47,53,75	5	64	6,42
1/3	557,663,711	8	512	7,78
1/2	53,75	5	64	6,02

1/2	61,73	5	64	6,02
1/2	71,73	5	64	6,02
1/2	133,171	6	128	6,99
1/2	247,371	7	256	6,99

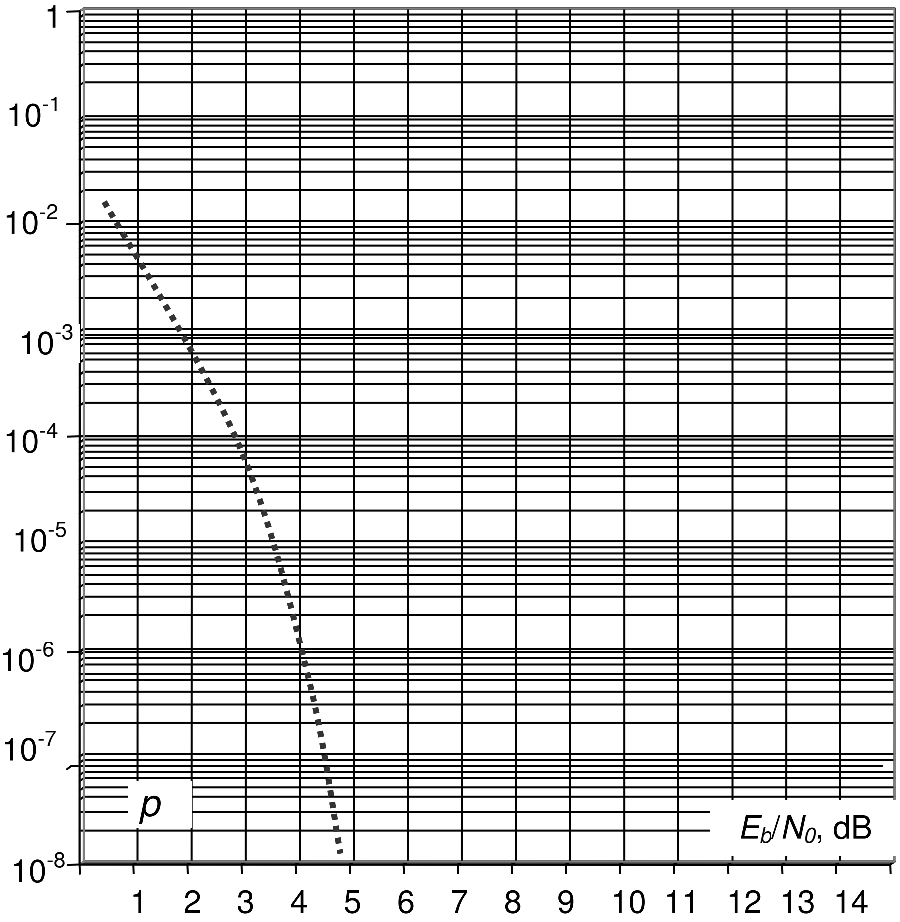


Figure B.1– Decoding noise immunity of a code(133,171)

TableB.2–***Input data for the course work***

Variant number for an elaborating of course work should correspond to the number of a student surname in the of the academic group journal							
Variant number	Transmitted digital information	. Ratio *S/N* h^2_0 (dB)	Modulation method	Source productivity R_{source} (kbit/s)	Channel frequency band F_{ch} (kHz)	Bit error probability p	Lattice Complexity W
1	CD	4,0	PhM-4	64	80	10^{-6}	150
2	DT	5,0	PhM-4	16	25	10^{-4}	160
3	DB	6,0	PhM-2	256	800	10^{-5}	170
4	CD	6,5	PhM-2	64	200	10^{-6}	180

5	DT	4,0	PhM-4	16	25	10^{-4}	250
6	DB	7,0	PhM-4	128	200	10^{-5}	350
7	DTV	5,0	PhM-2	2400	7000	10^{-8}	560
8	CD	6,0	PhM-4	32	50	10^{-6}	200
9	DT	5,0	PhM-2	24	70	10^{-4}	300
10	DB	4,5	PhM-4	256	400	10^{-5}	250
11	DTV	5,5	PhM-2	3000	1200	10^{-8}	550
12	CD	4,0	PhM-4	48	70	10^{-6}	150
13	DT	4,0	PhM-4	32	50	10^{-4}	250
14	DB	5,0	PhM-2	256	800	10^{-5}	300
15	DTV	4,0	PhM-4	4500	1300	10^{-9}	550
16	CD	7,0	PhM-4	56	90	10^{-6}	150
17	DT	5,0	PhM-2	24	70	10^{-4}	160
18	DB	4,5	PhM-4	256	400	10^{-5}	200
19	DTV	5,5	PhM-4	5000	1400	10^{-9}	550
20	CD	6,0	ФМ-2	64	200	10^{-6}	150
21	DT	7,5	PhM-4	32	400	10^{-4}	250
23	DT	6,5	ФМ-4	16	50	10^{-5}	150
24	CD	6,0	ФМ-4	64	150	10^{-6}	150
25	DT	4,5	PhM-2	16	25	10^{-4}	200
26	DTV	5,0	PhM-2	6000	16000	10^{-9}	550
27	DB	6,0	PhM-4	384	600	10^{-5}	250
28	CD	4,5	PhM-4	56	100	10^{-6}	150
29	DT	5,0	PhM-2	16	50	10^{-5}	250
30	DTV	5,5	PhM-2	5500	32000	10^{-9}	560
31	DT	4,5	PhM-4	64	200	10^{-5}	150
32	CD	5,0	PhM-4	64	300	10^{-5}	250

References

1.Стеклов В.К., Беркман Л.Н. Теорія електричного зв’язку: Підручник для ВНЗ. Під редакцією В.К. Стеклова. – К.: Техніка, 2006.–550 с.

2.Банкет В.Л.,П.В.Иващенко,А.Э. Геер. Цифровые методы передачи информации в спутниковых системах связи:Учебн. пособ.
– Одесса:УГАС,1996.–180 с.

3.Банкет В.Л.Дискретная математика в задачах теории цифровой связи: Учеб.пособие.– Одесса: ОНАС. 2008.–118 с.

4.Питерсон У.,Уэлдон Э.Коды, исправляющие ошибки./Пер. с англ.под ред.Р.Л. Добрушина.М.:Мир,1976.–594 с.

5.J.G.Proakis.Digital communications.–2nd ed.McGraw Hill Book Company.N.Y.1989 –905 p.

6.Скляр Б. Цифровая связь. Теоретические основы и практическое применение Изд. 2-е: Пер. с англ.–М.: Издательский дом «Вильямс»,2003. – 1104 с.